OR

SW14

Worcestershire, UK

Proceedings

Edited by
Dr Benny Tjahjono,
Dr Cathal Heavey,
and Dr Stephan Onggo

1-2 April 2014

Abbey Hotel, Worcestershire

THE OR SOCIETY

in association with

informs Simulation Society

SCS

The OR Society gratefully acknowledges the support of the following sponsors

LANNER the value of knowing

palgrave macmillan

SIMUL8 CORPORATION

dse CONSULTING

saker solutions

[dstl]

PROCEEDINGS

SW14

8TH SIMULATION WORKSHOP

Abbey Hotel, Worcestershire, UK

1 – 2 April 2014

TABLE OF CONTENTS

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

MODEL BUILDING IN SYSTEM DYNAMICS AND DISCRETE-EVENT SIMULATION: A COMPARISON OF ANALYSTS' LANGUAGE

Dr. Roger McHaney

Kansas State University
Manhattan, KS 66506, USA
mchaney@ksu.edu

Dr. Antuela A. Tako

Loughborough University
Loughborough, LE11 3TU, UK
A.Takou@lboro.ac.uk

Dr. Stewart Robinson

Loughborough University
Loughborough, LE11 3TU, UK
S.L.Robinson@lboro.ac.uk

ABSTRACT

This article presents an analysis derived from empirical data collected during a study on the differences in System Dynamics (SD) and Discrete-event Simulation (DES) model building. The language usage of 10 expert modellers (5 SD and 5 DES), who provided a narrative of their actions while building prison simulation models, formed the study's basis. The transcripts were analysed using Linguistic Inquiry and Word Count (LIWC), a text analysis software program which calculates the degree to which people use different word categories across a wide array of areas, and then loads the results on more than 70 different dimensions. The purpose of this study was to determine whether distinctive features of language could be discerned in the language usage of SD and DES experts. Theoretical differences were hypothesized based on prior studies. Results indicated language usage was consistent with hypothesized characteristics of SD and DES and further validated earlier studies.

Keywords: System Dynamics, Discrete-Event Simulation, Comparison, Simulation, Model Building, Linguistic Inquiry and Word Count, Simulation Success

1 INTRODUCTION

Discrete-Event Simulation (DES) and System Dynamics (SD) are two approaches used to develop models for a wide range of applications in operations research and other areas. While both approaches have similar motivations, they evolved separately, with little overlap between users of each technique. A recent study on the modelling process by Tako and Robinson (2010) showed analysts using DES follow a more linear progression with more focus on model coding, verification, and validation, while analysts using SD are more concerned with conceptual modelling. Prior to that research, few studies focused on differences and similarities between the two techniques. Most articles reporting to compare DES and SD often used the authors' personal opinions and experiences as a framework for analysis (Brailsford and Hilton, 2001; Morecroft and Robinson, 2005).

This study extends prior work as a basis for further investigating similarities and differences between approaches used by DES and SD analysts. Additionally, this study views those differences from a unique perspective recently developed by social psychologists. We hope to better understand the cognitive processes used by SD and DES development through linguistic analysis and determine if various characteristics of their modelling approaches appears to manifest in dimensions of language used to describe their process.

Several hypotheses were formulated based on theoretical work describing computer simulation success factors in the DES area (McHaney and Cronan, 1998) and other work in DES and SD. These hypotheses were used to describe expected similarities and differences in the way analysts describe their processes. Further analysis was conducted to determine if broad stroke differences in language complexity and use existed, and whether these differences could be attributed to the technique being employed by the analysts.

Therefore, this paper presents an empirical study on the comparison of SD and DES modelling by investigating linguistic similarities and differences through a theoretical lenses provided by prior research. As in the original study, we acknowledge key fundamental differences exist between SD and DES simulation models and respective software use. However, the primary objective of this article is to provide empirical evidence on the differences and similarities in the *cognitive* processes during SD and DES modelling as manifested in language use.

The current study uses data collected during laboratory experiments, where ten expert modellers (five SD and five DES modellers) were observed while building simulation models of a UK prison population. A qualitative research technique called Verbal Protocol Analysis (VPA), where the expert modellers are asked to speak aloud their thoughts while they are undertaking a simulation modelling exercise, was used to collect the data. All material analysed was derived during live observation of these expert modellers and was transcribed to text files.

The aim of this study is to provide further quantitative comparison of the simulation modelling processes followed by SD and DES modellers. The results provide insight into this body of knowledge and continue to bring these two fields of simulation closer, with a goal of eventually creating a common basis of understanding.

The article is outlined as follows: it starts with a review of the existing literature on the comparison of DES and SD, followed by a description of the prior study undertaken together with sections where Verbal Protocol Analysis (VPA), Linguistic Inquiry and Word Count and other techniques are described. Then, quantitative analyses based on observations from the ten modelling sessions, are presented. Finally, the limitations and the main findings of the current study are discussed. Conclusions summarize the outcomes.

1.1 SD and DES Comparisons in the Literature

Existing literature comparing SD and DES is limited but growing (Morecroft and Robinson, 2005). Until recently, most comparisons were not empirical but instead primarily focused on authors' views of the two approaches with a tendency toward bias related to foundational roots and author expertise (Brailsford and Hilton, 2001). A historic lack of dialog between the two modelling communities appears to have resulted in the independent growth of similar approaches and techniques (Sweetser, 1999; Lane, 2000). Recently, more academics and practitioners appear to be open to sharing insights and learning from their peers with different backgrounds. Morecroft and Robinson (2005) explore this and suggest the two modelling approaches are different, but the application of both can yield complementary insights. Often the choice of an approach is based on prior experiences rather than choosing a more appropriate tool.

In general, SD models are an aspect of systems theory used to understand the behaviour of complex systems over time. SD is both a methodology and mathematically based modelling technique that can help define, analyse, and illuminate these systems through the use of internal feedback loops and small time delays (Δt) that dynamically affect the behaviour of the entire system (Forrester, 1961). An SD approach to representing complexity involves the use of feedback loops, stocks, and flows which can help explain nonlinearity. SD models represent simultaneity and causation through interrelated variables that are updated in small time increments. SD effectively represents time delays and complex relationships that produce system behaviours that cannot be determined by viewing individual system components alone. SD is holistic and well suited to strategic decision making.

Discrete Event Simulation models, in contrast, are commonly used by engineers, business analysts, and design specialists to aid in the decision-making process, particularly at operational or tactical levels. DES is distinguished by a particular characteristic-the passage of blocks of time during which no changes to the system state occur. DES relies on entity arrivals and service completions and resulting impact on a network of queues. The resulting interaction between constrained resources dynamically drives a model. Events take place instantaneously in discrete steps (McHaney, 1991).

1.2 Differences in SD and DES

A primary difference between SD and DES relates to entity representation. In SD, entities are a continuous quantity. In contrast, DES objects are individually represented and endowed with specific characteristics or attributes. DES entities have a location and can be tracked through a system often represented as a block diagram. In another difference, DES state changes occur at discrete times and SD state changes are continuous. SD is generally deterministic with variables that usually represent average values while DES models are stochastic and rely on statistical distributions to drive events with a degree of randomness.

Brailsford and Hilton (2001) suggest that DES models are not readily transparent to decision makers. Although animation and reporting techniques can provide useful insights, the logic is often deeply embedded in the model structure. This may require having a methodology to extract the logic and transfer it to a team responsible for system implementation (McHaney, 1988). SD models are often used for training purposes with logic and processes being more visible (Lane, 2000). However, SD models do not offer animation requiring the user to rely on graphs and numerical displays (Sweetser, 1999).

Table 1 provides a detailed view of the differences together with a categorization scheme that shows how these differences may impact the way analysts view SD and DES. These differences are used to explain why various dimensions in the expert modellers' descriptions manifest for the two techniques.

Table 1 – *Differences in DES and SD Modelling*

Problem Structure/Development Philosophy			
DES	SD	Category	References
Operational	Strategic	Best Problems for Technique	Lane (2000)
Stochastic	Deterministic	Model Approach	Brailsford and Hilton (2000); Coyle (1985); Morecroft and Robinson (2005)
Discrete	Continuous	State Changes	Brailsford and Hilton (2000)
Analytic; emphasis on detail complexity	Holistic; emphasis on dynamic complexity	Perspective	Lane (2000)
Detail Level	Aggregate Level; Big picture	Model Scope	Robinson (2004); Pidd (2004)
No agreed standard	Standard Structures	Diagramming Format	Morecroft and Robinson (2005)
Inputs			
DES	SD	Category	References
Activity durations from probability distributions	Time in reservoir modeled with limited flexibility	System Activity	Brailsford and Hilton (2000)
Individual entities, attributes, decision and events	Homogenised entities, continuous policy pressures, and emergent behaviour	Resolution	Lane (2000)
Features randomness prominently.	Stochastic variability subsumed into an appropriate delay; model structure driven	Randomness	Coyle (1985); Morecroft and Robinson (2005)
Primarily numerical with some judgmental elements	Broadly drawn	Data Sources	Lane (2000)

Internal Processes			
DES	SD	Category	References
Many linear	Many non-linear	Relationships	Morecroft and Robinson (2005)
Growth/decay represented as random with discrete steps	Growth/decay modeled as exponential or s-shaped	Relationships	Morecroft and Robinson (2005)
Standard structures rarely exist	Recurring structures	Modelling Structures	Morecroft and Robinson (2005)
Open-process structure	Closed-loop structure with feedback	Problem Structure	Coyle (1985)
Networks of queues and activities	Series of stocks and flows	System Representation	Brailsford and Hilton (2000)
Objects are distinct individuals with characteristics	Objects are continuous quantity	Object Representation	Brailsford and Hilton (2000)
Unequal timesteps; when “something happens”	Finely-sliced time steps of equal duration	Time	Brailsford and Hilton (2000)
Feedback			
DES	SD	Category	References
Feedback held in model logic	Explicit feedback	Placement	Mak (1993)
Implicit	Explicit	Transparency	Morecroft and Robinson (2005)
Not emphasised	Vital	Importance	Morecroft and Robinson (2005)
Output			
DES	SD	Category	References
Opaque/dark grey box	Transparent/fuzzy glass box	Client View	Lane (2000)
Statistically valid estimates of system performance measures	Results considered source of understanding reasons for changes in system performance	Outputs	Sweetser (1999); Brailsford and Hilton (2001); McHaney (1991); Morecroft and Robinson (2005)
Usage			
DES	SD	Category	References
Decision Makers	Bounded, rational policy implementers	Human Component	Lane (2000)
Point predictions and detailed performance measures across a range of parameters, decision rules and scenarios (Detailed)	Understanding of structural source of behaviour modes, location of key performance indicators and effective policy levers (Structural)	Model Usage Scope	Lane (2000)
‘What-if’ experimentation used to investigate various scenarios	Study interaction of control policies, exogenous events and feedback structure	Question Types	Mak (1993)

Performance over time subject to internal and external conditions	Performance over time subject to internal feedback structure	Time Related	Morecroft and Robinson (2005)
Experts and non-experts	System users learning from use	User Characteristics	Sweetser (1999); Forrester (1961);Robinson (2004); Morecroft and Sterman (1994)

Despite a variety of differences, a number of commonalities have been noted in the literature. For instance, both simulation approaches fundamentally seek to provide a better understanding of how systems behave over time under varying conditions. Supporting this view are observations of end user perceptions that SD and DES models are not significantly different. Since both approaches (and various other modelling techniques) are concerned with building accurate, representational models, yielding results that lead to confident decisions, it has been suggested that both DES and SD can represent reality with equal validity (Akkermans, 1995). This study does not seek to promote one technique as superior. Instead, understanding the strengths and weaknesses of both can help strengthen both approaches and ensure the correct approach is matched to appropriate problems.

1.3 Study and Analysis Goal

The primary objective of this research is to investigate whether differences hypothesized to exist between the DES and SD approaches to simulation are reflected in the language used to describe construction of specific models in each area. An empirical language analysis of DES and SD model building processes was conducted and used to provide insights for both areas. The insights then were used to build support for subjective statements made by a variety of experts and for a prior empirical study (Tako and Robinson, 2010).

The current study used data collected during research by Tako and Robinson (2010). In the original study, empirical data was used to compare DES and SD approaches used by expert simulation modellers. Tako and Robinson (2010) compared the DES and SD model building process as narrated by these experts, with Verbal Protocol Analysis (VPA), a qualitative research method (Green, 1998). In the current study, the same data was further analysed using an additional technique: Linguistic Inquiry and Word Count (LIWC) text analysis (Pennebaker, Francis and Booth, 2001).

To achieve our primary objective, we empirically compare the self-described behaviour of expert modellers while each undertakes a similar simulation modelling task. We believe that analysts using DES and SD approaches think differently during the model building process. It is expected that language selection across various thresholds, as measured by LIWC dimensions will reflect this difference in thinking. LIWC uses both qualitative and quantitative text analysis techniques to identify hypothesized characteristics common to each text sample and then these characteristics are statistically tested for significant differences. The resulting differences will provide insight regarding differences in DES and SD.

1.4 Prior Study and Data Collection

As stated earlier, the data used for analysis in the current study was collected by Tako and Robinson (2010). In their original study, they used a case study based on UK prisons. The original study had an overall objective to empirically compare the behaviour of expert modellers while undertaking a simulation modelling task. They "conducted a quantitative comparison of expert modellers' thinking, looking at the process they think about while building simulation models." (Tako and Robinson, 2010, p. 786).

For this case, prisoners enter prison initially as first time offenders and then may return back to prison as recidivists. This general scenario was represented by simple simulation models, using both DES and SD techniques as constructed by known experts in each area. The case idea was based on prior modelling work in DES (Kwak, Kuzdrall and Schniederjans, 1984; Cox, Harrison and Dightman, 1978; Korporaal, Ridder and Kloprogge, 2000); and in SD (Bard, 1978). The specific UK prison population case study used in this research was based on Grove, Macleod and Godfrey (1998).

As originally described in Tako and Robinson (2010, p. 787), the case study introduces the prison population problem with a focus on the issue of overcrowding. The case contained descriptions of reasons for the problem and suggested impacts caused by this problem. The facts provided to the modelling experts were loosely based on reality, but contained adaptions tailored to collecting data for the research.

Two prisoner groups were examined in the scenario to be modelled: petty criminals and serious offenders. The prison systems starts with an existing population of prisoners, set at 76,000. First time offenders enter the system and were sentenced based on their offence type. Petty criminals entered the system at a higher rate averaging three thousand per year while six hundred and fifty serious offenders entered the system per year. Petty offenders averaged a five year sentence while serious offenders were given an average sentence of twenty years. Following time served, offenders in both categories were released. A percent of the released prisoners reoffended and were returned to prison after an average of two years. Petty prisoners were more likely to be reoffenders, but the percent of recidivism was not provided to the expert modellers who either made their own assumptions or could ask for more data.

The modellers considered two possible scenarios: (1) increase current prison capacity thereby facilitating more rigid rules; or (2) reduce the size of the prison population by providing alternatives to jail and enhancements to social support provided to prisoners. The modelling experts were asked to create a model for use in decision support by policy makers. .

1.5 Data Collection

Data used for analysis was collected using the qualitative research method, Verbal Protocol Analysis (VPA) . Using this approach requires test subjects to 'think aloud' during a problem-solving exercise, with a particular focus on problem-solving and decision making instances. The technique uses participants' verbal output to provide insights regarding thought processes and the internal structure of cognitive activity (Ericsson and Simon, 1984).

Tako and Robinson (2010) suggested VPA would be an effective method for the comparison of the SD and DES model building process due to the "richness of information and the live accounts it provides on the experts' modelling process." (Tako and Robinson, 2010, p. 787) Other options were considered and ruled out, including observation of real simulation development projects (determined to suffer from elongated time scales and difficulty in comparing dissimilar modelling situations), and interviews with modelling experts (may not fully capture the development process). VPA, on the other hand, offered elements of control and the ability to capture a complete representation of the modelling process.

VPA and protocol analysis in general has limitations. According to Willemain (1995), experts under observation may act differently resulting in incomplete data collection. There is a risk that the experts fail to verbalize all their cognitive processes or elements of interaction and collaboration might be missing. To help minimize the limitations, verbalization exercises were conducted prior to beginning data collection.

Ten simulation experts were involved in the VBA sessions with 5 from DES and 5 from SD (Tako and Robinson, 2010, p. 788). The study's subjects were given the prison population case study and the VPA session was started. The experts were asked to build models using their preferred simulation approach. During the entire process, experts were encouraged to verbalize their thoughts. The verbal protocols were recorded on audio tape and then transcribed.

1.6 Procedure for Current Study

The collected data was placed into text files and checked for accuracy and steps were taken to eliminate data problems as recommended by LIWC's developers (Pennebaker, Booth and Francis, 2007). This included fixing misspellings, replacing abbreviations, contractions, sentence end markers, and addressing other commonly identified potential problems. Following this process, each text file was analysed using the LIWC program.

LIWC is computer software designed by Pennebaker, Booth, and Francis (2007), experts in social psychology. LIWC categorizes and counts words in psychologically meaningful categories so inferences and comparisons can be made. Empirical results from uses of LIWC have demonstrated the ability to support meaning in a wide variety of situations (Tausczik and Pennebaker, 2010). Past uses have analysed text messages, emails, speeches, poems, conversations, and written material of many types. LIWC typically is used to determine the degree analysed text language exhibits characteristics consistent with more than seventy language dimensions including various emotional states, self-reference, attention focus, social relationships, thinking styles, individual differences, perception, causality, and so forth. Each of the language categories has been rated independently by expert judges and subjected to extensive validation and psychometric stability testing (Pennebaker and King, 1999; Pennebaker, 1997). For example, the word "sobbed" would be categorized into four dimensions: past tense verb, sadness, overall affect, and negative emotion.

The language used by DES and SD simulation experts during descriptions of model development is suited to a linguistic analysis due to the differences in perspective, model scope, model uses, and other factors as described in Table 1. Further, to support hypothesized differences in cognitive processing engaged during building models in their respective domain, simulation experts will be more likely to use language that correlates with unique aspects of their approach. For instance, analysts using a SD approach will be more likely to use holistic, policy-level and strategic word choices. Analysts using a DES approach will be more likely to focus on details, analysis, and individualistic words. Table 2 provides a list of LIWC categories hypothesized as meaningful to this study. Reasons for the meaning are also provided.

Table 2 - *LIWC Categories Theoretically Relevant to SD/DES Comparison*

LIWC Dimension	**Hypothesized Relationship**	**Description**
H1: Words > 6 Letters	SD > DES	Correlates with social class and SD users typically speak the language of executives while DES is more operational.
H2: Personal Pronouns	DES > SD	This is consistent with theory that indicates DES is more operational and involves individual representations of entities (workers, prisoners, et cetera).
H3: Second Person	SD > DES	Use of words related to elevated social status. SD users speak the language of executives.
H4: Affective Processes	DES > SD	This is consistent with theory that indicates DES is more operational and involves individuals and might therefore be more likely to consider human actions more and be more emotional.
H5: Insight	SD > DES	SD looks at policies and higher level decisions which lead to insight. It also is frequently used in decision making.
H6: Causation	SD > DES	SD provides a way to see the big picture which might yield insight into cause and effect.
H7: Tentative	DES > SD	DES is stochastic and susceptible to disruption and inclusion of 'long tail' effects.
H8: Discrepancy	DES > SD	DES is likely to have discrepancies caused by its stochastic nature
H9: Certainty	SD > DES	SD is less dependent on stochastic processes.
H10: Inhibition	DES > SD	This dimension indicates some uncertainty and correlates with maybe, perhaps, and guessing. Since DES is more detailed and requires a finer resolution, more items come under close scrutiny.
H11: Inclusive	SD > DES	Feedback is incorporated into SD.
H12: Exclusive	SD>DES	DES often embeds and obscures relationships while SD makes them more obvious.
H13: Perception	DES > SD	Perception is related to observing, hearing, and feeling. This dimension supports theorists that suggest DES requires more detail in the development stages and offers more a visual, perceptual output (e.g. animation).
H14: Seeing	DES > SD	Animation is often used in DES.
H15: Relativity	DES > SD	DES has a detailed nature and reliance on physical relationships, time, and, in some cases, distance.
H16: Motion	DES > SD	DES has a detailed nature and reliance on physical relationships, time, and motion.
H17: Time	DES > SD	DES is time-based to a higher extent than SD.
H18: Space	DES > SD	DES has a detailed nature and reliance on physical relationships, time, and, in space (e.g. layouts in factory design and so forth).

H19: Work	DES > SD	DES is operational and relies more on the human component
H20: Achievement	SD > DES	SD is concerned with higher level decision making and policy which often includes goal setting and achievement.
H21: Question	DES > SD	DES is more detail oriented and this should be reflected in the language
H22: Emotion	DES > SD	DES is operational and more likely to include emotions in decisions and human process modelling.

2 METHODS

A total of 10 simulation development projects were observed. The subjects generating transcripts for analysis were all simulation experts. To maintain confidentiality participants' names are not disclosed. Instead, transcribed sessions were placed into text files designated with the symbol DES or SD and identification number according to the technique used. All expert participants use the appropriate modelling technique as part of their job functions in their corporate lives and performed their tasks using popular simulation software tools. Each participant was employed by an established simulation software or consultancy company based in the UK. Each participant held at least a masters' degree in engineering, computer science, operational research, or business administration with additional specialized simulation training. Their industry experience in modelling ranges from 4 years to 20 years.

2.1 Analysis

The results provided by LIWC were aggregated by simulation method using an Excel spreadsheet. This created two sets of data that were analysed according to the language dimensions and other measures provided by the software. Independent-samples t-tests were conducted the results of each language dimension provide by LIWC. Significant differences were noted in many of the categories.

2.2 Results

An initial review of the data provided information regarding characteristics of the data set. Cleaned transcripts for DES ranged from 3087 to 8355 words with an average size of approximately 5394 words. SD transcripts ranged from 2858 to 4772 words with an average size of 3801. This finding is consistent with the literature which suggests that DES focuses more on details and SD more on an overview. Words per sentence were nearly identical with DES averaging 17.5 and SD averaging 17.6 indicating comparable cognitive complexity and fluency levels (Tausczik and Pennebaker, 2009, p. 39).

Table 3 summarizes the significant results and Table 4 provides a summary with regard to the original hypotheses. The first measure demonstrating a significant difference was the percent of all words exceeding six letters in length. There was a significant difference in the scores for H1: Words > 6 Letters dimension between DES (M=12.05, SD=1.09) and SD (M=15.81, SD=1.80) p = .004. According to Tausczik and Pennebaker, 2009, p. 39) this measure has psychological correlation with social class. This is consistent with the theoretical view that SD is typically used at a strategic level and DES is used in a more operational way. It is expected that SD would be closely associated with the language typically used by upper executives.

There was a significant difference in the scores for H2: Personal Pronouns dimension: DES (M=8.45, SD=1.12) and SD (M=6.95, SD=0.67) p = .033. According to Tausczik and Pennebaker, 2009, p. 40) this measure has psychological correlates with personal involvement and social interaction. This is consistent with theory that indicates DES is more operational and often involves individual representations of entities (workers, prisoners, et cetera).

H5: Insight is another dimension with a significant difference between DES and SD. The scores for this dimension DES (M=1.55, SD=0.44) and SD (M=2.08, SD=0.43) p = .089 support the literature-based viewpoints that SD offers learning potential and understanding. According to Tausczik and Pennebaker (2009, p. 41) this measure has psychological correlation with knowing, thinking and considering.

The dimension for H10: Inhibition shows a significant difference. Inhibition indicates constrains, stops and problems. This dimension indicates some uncertainty and correlates with *maybe*, *perhaps*, and *guessing* (Tausczik and Pennebaker, 2009, p. 41). There was a significant difference in the scores for this dimension: DES (M=.414, SD=0.12) and SD (M=0.27, SD=0.09) p = .062. This difference perhaps can be attributed to the theoretically indicated difference in modelling approaches. Since DES is more detailed and requires a finer resolution, more items come under close scrutiny. Additionally, DES requires using statistics to interpret the outcomes and this might require additional thought being given to experimental design.

H13: Perception dimension also shows a significant difference. Perception is related to *observing*, *hearing*, and *feeling* (Tausczik and Pennebaker, 2009, p. 42). This dimension supports theorists that suggest DES requires more detail in the development stages and offers more visual, perceptual output (e.g. animation). This dimension was significant DES (M=1.69, SD=0.50) and SD (M=0.88, SD=0.26) p = .011. Supporting this viewpoint is the H14: Seeing dimension DES (M=.89, SD=0.13) and SD (M=0.47, SD=0.15) p = .001.

Two more dimensions found to be significant are for H15: Relativity and H16: Motion (Tausczik and Pennebaker (2009, p. 42). Both of these dimensions deal with physical perspective and load higher for DES. This makes theoretical sense because of its detailed nature and reliance on physical relationships, time, and, in some cases, distance. H15: Relativity dimension had DES (M=15.59, SD=2.60) and SD (M=12.63, SD=1.66) p = .064 differences. The motion dimension was DES (M=3.98, SD=0.55) and SD (M=2.53, SD=0.35) p = .001.

The final significant dimension deals with questions: H21: Question. DES transcripts contain both more questions which appear to be theoretically consistent with a DES requirement for more detail. The QMark dimension's scores were DES (M=0.53, SD=0.26) and SD (M=.025, SD=0.05) p = .043.

Table 3 - *Significant Results of LIWC Analysis*

LIWC Dimension	**DES**	**SD**	**p-Value**	**Theoretical Explanation**
H1: Words > 6 Letters	M=12.05SD =1.09	M=15.81 SD=1.80	.004	Psychological correlate with social class. This is consistent with the theoretical view that SD is typically used at a strategic level and DES is used in a more operational way. It is expected that SD would be closely associated with the language typically used by upper executives.
H2: Personal Pronouns	M=8.45 SD=1.12	M=6.95 SD=0.67	.033	Psychological correlates with personal involvement and social interaction. This is consistent with theory that indicates DES is more operational and often involves individual representations of entities (workers, prisoners, et cetera).
H5: Insight	M=1.55 SD=0.44	M=2.08 SD=0.43	.089	Insight is another dimension with a significant difference between DES and SD. The scores on this dimension support the literature-based viewpoints that SD offers learning potential and understanding. According to Tausczik and Pennebaker (2009, p. 41) this measure has psychological correlates with knowing, thinking and considering.
H10: Inhibition	M=.414, SD=0.12	M=0.27, SD=0.09	.062	Inhibition indicates constrains, stops and problems. This dimension indicates some uncertainty and correlates with maybe, perhaps, and guessing (Tausczik and Pennebaker, 2009, p. 41). Difference perhaps can be attributed to the theoretically indicated difference in modelling approaches. Since DES is more detailed and requires a finer resolution, more items come under close scrutiny. Additionally, DES requires using statistics to interpret the outcomes and this might require a higher degree of thought being given to experimental design.

H13: Perception	M=1.69, SD=0.50	M=0.88, SD=0.26	.011	A perception dimension also shows a significant difference. Perception is related to observing, hearing, and feeling (Tausczik and Pennebaker, 2009, p. 42). This dimension supports theorists that suggest DES requires more detail in the development stages and offers more a visual, perceptual output (e.g. animation).
H14: Seeing	M=.89, SD=0.13	M=0.47, SD=0.15	.001	This dimension supports the visual component of DES and how outputs are viewed in visually stimulating ways.
H15: Relativity	M=15.59, SD=2.60	M=12.63, SD=1.66	.064	Deals with physical perspective and loads higher for DES. This makes theoretical sense because of its detailed nature and reliance on physical relationships, time, and, in some cases, distance.
H16: Motion	M=3.98, SD=0.55	M=2.53, SD=0.35	.001	Loads reasons similar to relativity
H21: Question	M=0.53, SD=0.26	M=.025, SD=0.05	.043	Deals with questions. DES transcripts contain both more questions (theoretically consistent with a requirement for more detail)

Table 4 - *Summary of Results Related to Original Hypotheses*

LIWC Dimension	Hypothesized Relationship	Result
H1: Words > 6 Letters	SD > DES	Supported (p=.004)
H2: Personal Pronouns	DES > SD	Supported (p=.033)
H3: Second Person	SD > DES	Not Supported
H4: Affective Processes	DES > SD	Not Supported
H5: Insight	SD > DES	Supported (p=.089)
H6: Causation	SD > DES	Not Supported
H7: Tentative	DES > SD	Not Supported
H8: Discrepancy	DES > SD	Not Supported
H9: Certainty	SD > DES	Not Supported
H10: Inhibition	DES > SD	Supported (p=.062)
H11: Inclusive	SD > DES	Not Supported
H12: Exclusive	SD > DES	Not Supported
H13: Perception	DES > SD	Supported (p=.011)
H14: Seeing	DES > SD	Supported (p=.001)
H15: Relativity	DES > SD	Supported (p=.064)
H16: Motion	DES > SD	Supported (p=.001)
H17: Time	DES > SD	Not Supported
H18: Space	DES > SD	Not Supported
H19: Work	DES > SD	Not Supported
H20: Achievement	SD > DES	Not Supported
H21: Question	DES > SD	Supported (p=.043)
H22: Emotion	DES > SD	Not Supported

3 DISCUSSION

In general, the results indicate that certain cognitive processes are more likely to occur in one modelling approach (DES vs. SD) than the other which is consistent with prior research and academic opinions. These differences were manifested in language dimensions specifically measured by LIWC. Additionally, significant findings from this study provide validation for prior empirical study in the area. Specifically, Tako and Robinson (2010) concluded the *behaviour* of expert analysts using SD and DES approaches was different when building simulation models. The contribution of their paper related to a comparison of SD and DES model building process and the actions taken by the experts. The current research supports this finding by concluding the *cognitive processes* of these experts, as manifested in language dimensions, also was different for analysts using SD and DES approaches.

Word choices reflecting underlying cognitive processes indicate that analysts using an SD approach tend to be more strategic and correlate with LIWC dimensions consistent with language use in an executive rather than operational or tactical setting (H1). This is further supported by the finding that DES use focuses more on personal involvement and individual representations of entities in the model (H2). These two significant differences provide empirical evidence to support theoretical beliefs that SD is more strategic and DES often is more operational.

Another area of difference was the way insight (H5) was represented in language use. Analysts using SD approaches were theorized as using their models more for learning and education potential when dealing with clients. LIWC's insight dimension supported this view.

Additionally, according to theory, experts using DES are more concerned with uncertainty, detail, attempting to interpret outcomes with statistics, and using stochastic measures as inputs. These views were supported. For instance, in H10 LIWC loaded higher for experts using DES and indicated more cognitive processing dealt with uncertainty, constraints, and problem details. Other support came from H13 and H14 which suggest experts using DES are more likely to require detail in development and put more emphasis on visual components of modelling. Again, this is consistent with prevailing theory regarding the modelling approaches.

Further, experts using DES tend to be more focused on physical dimensions and detailed descriptions of the systems being modelled. The LIWC dimension for relativity (H15) loads higher and supports theory from the literature in this area. Another related dimension, motion (H16) also loads higher reflecting the physical nature of DES modelling practice. Finally, experts using DES tend to spend more cognitive processing on questions as indicated by a significantly higher loading on the question dimension (H21).

While these findings are interesting, some surprises were also encountered. For instance, in spite of prevailing thought that DES use is more externally focused and SD use relies more on internal feedback loops, no significant difference in LIWC dimensions for these items were detected (H11 and H12). Perhaps the reasons for this are related to constraints of the prison model selected for analysis. Likewise dimensions for causality, certainty, and discrepancy did not show any significant difference in language use by experts in either area.

3.1 Limitations

This study does have several limitations which should be considered. First, the data are derived from artificial laboratory settings, where the modellers knew they were under observation and were under the pressure of time and of acting like experts. The task itself could also be considered a limitation and perhaps even related to reasons some hypothesized outcomes were found not significant. The prison model was relatively simple and structured to ensure completion of the exercise in a limited amount time. Additionally, the subject was less physical in manifestation than a manufacturing or industrial model, typical for DES, might have been. Another limitation is the sample size. A larger group of experts might have provided more representative results. More experts and multiple cases could have provided a richer set of data for analysis. Another limitation is the LIWC software. While it checks for over 70 validated dimensions, many others may exist that are not being examined. Likewise, specialty language use might not be exactly categorized as expected. Finally, the alignment of the LIWC dimensions with the hypothesized differences between SD and DES use is open to some subjectivity, and so we must be cautious about the interpretation of the results.

3.2 Conclusions

This study validates an empirical study that compared SD and DES model building based on data gained from experimental exercises involving expert modellers. The primary results of the study were consistent with theory and academic conjecture and provide knowledge that could be used in the better selection of an appropriate modelling method. In addition, this article uses LIWC as a technique that can provide knowledge in the operational research area by examining underlying cognitive process through language use. The ideas used here can be implemented in other studies where written and verbal qualitative data is available.

REFERENCES

Akkermans, H. (1995). Developing a logistics strategy through participative business modelling. *International Journal of Operations & Production Management*, *15*(11), pp. 100-112

Bard, J. F. (1978). "The Use of Simulation in Criminal Justice Policy Evaluation" *Journal of Criminal*

Justice 6(2): pp. 99-116.
Brailsford, S. and N. Hilton (2001). “A comparison of discrete event simulation and system dynamics for modelling healthcare systems.” *Proceedings of the 26th meeting of the ORAHS Working Group 2000*, Glasgow Caledonian University: Glasgow, Scotland.
Coyle, R. G. (1985). "Representing discrete events in system dynamics models: a theoretical application to modelling coal production" *Journal of Operational Research Society* 36(4): pp. 307-318.
Cox, G.B., P. Harrison, and C.R. Dightman (1978). Computer simulation of adult sentencing proposals. *Evaluation and Program Planning*, 1(4): pp. 297-308.
Ericsson, K. A. and H. A. Simon (1984). *Protocol analysis: verbal reports as data*. The MIT Press.
Forrester, J.W. (1961). *Industrial Dynamics*. Massachusetts Institute of Technology Press.
Green, A. (1998). *Verbal protocol analysis in language testing research: a handbook*. Studies in Language Testing, 5. Cambridge: Cambridge University Press.
Grove, P., J. Macleod, and D. Godfrey (1998). Forecasting the prison population. *OR Insight* 11(1): pp. 3-9.
Korporaal, R., A. Ridder, and P. Kloprogge (2000). An analytic model for capacity planning of prisons in the Netherlands. *Journal of the Operational Research Society*, 51(11): pp. 1228-1237.
Kwak, N.K., P.J. Kuzdrall, and M.J. Schniederjans, M. J. (1984). Felony case scheduling policies and continuances: A simulation study. *Socio-Economic Planning Sciences*, *18*(1): pp. 37-43.
Lane, D. C. (2000). “You just don't understand me: models of failure and success in the discourse between system dynamics and discrete event simulation,” *LSE OR*.
Mak, H.Y. (1993). System dynamics and discrete event simulation modelling. Department of Statistical and Mathematical Sciences. London, London School of Economics and Political Science.
McHaney, R. (1988). “Bridging the gap: transferring logic from a simulation into an actual system controller,” In *Proceedings of the 20th conference on Winter simulation* (WSC '88), Michael A. Abrams, Peter L. Haigh, and John C. Comfort (Eds.). ACM, New York, NY, USA: pp. 583-590.
McHaney, R. (1991). *Computer Simulation: A Practical Perspective*. Academic Press: San Diego.
McHaney, R. and T.P. Cronan (1998), “Computer Simulation Success: On the Use of the End-User Computing Satisfaction Instrument: A Comment,” *Decision Sciences*, *29*: pp. 525–535.
Morecroft, J.D.W. and S. Robinson (2005). “Explaining puzzling dynamics: comparing the use of system dynamics and discrete-event simulation,” *In Proceedings of the 23rd International Conference of the System Dynamics Society*: Boston.
Morecroft, J. and J. Sterman (1994). *Modelling for learning organizations*. Productivity Press: Portland, OR.
Pennebaker, J.W. (1997). Opening up: the healing power of expressing emotion. Guilford Press; New York.
Pennebaker, J.W., M.E. Francis, and R.J. Booth (2001). Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates*, 71.
Pennebaker, J.W. and L.A. King (1999). Linguistic styles: language use as an individual difference. *Journal of personality and social psychology* 77(6): pp. 1296–1312.
Pidd, M. (2004), *Systems modelling: theory and practice*. John Wiley & Sons.
Robinson, S. (2004). *Simulation: the practice of model development and use*. Wiley: Chichester.
Sweetser, A. (1999). “A comparison of system dynamics and discrete event simulation,” *In Proceedings of 17th International Conference of the System Dynamics Society and 5th Australian & New Zealand Systems Conference*: Wellington, New Zealand.
Tako, A.A. and S. Robinson (2009). “Comparing discrete-event simulation and system dynamics: users’ perceptions.” *Journal of the Operational Research Society*, 60 (3): pp. 296-312.
Tako, A.A. and S. Robinson (2010). “Model development in discrete-event simulation and system dynamics: An empirical study of expert modellers.” *European Journal of Operational Research* 207(2): pp. 784-794.
Tausczik, Y.R. and J.W. Pennebaker (2010). The psychological meaning of words: LIWC and

computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1): pp. 24-54.
Willemain, T.R. (1995). "Model formulation: what experts think about and when" *Operations Research* 43(6): pp. 916-932.

AUTHOR BIOGRAPHIES

ROGER McHANEY is a Professor of Management Information Systems and University Distinguished Teaching Scholar at Kansas State University, College of Business. He is the author of books on simulation, education technology and Web 2.0 applications. Prior to becoming an academic, he developed simulations primarily in manufacturing and materials handling. His current research includes understanding how simulation and modelling techniques can best be applied in business organizations, and data-driven simulations.

ANTUELA A. TAKO is a Lecturer in Operations Research at Loughborough University, UK. She holds a PhD in Simulation and an MSc in Management Science and Operational Research from the University of Warwick. Her research focuses on the comparison of simulation approaches (Discrete-Event Simulation and System Dynamics), facilitative and participative simulation modelling and conceptual modelling. She is an Associate member of the Operational Research Society (AORS), UK.

STEWART ROBINSON is Professor of Management Science and Associate Dean Research at Loughborough University, School of Business and Economics. Previously employed in simulation consultancy, he supported the use of simulation in companies throughout Europe and the rest of the world. He is author/co-author of five books on simulation. His research focuses on the practice of simulation model development and use. Key areas of interest are conceptual modelling, model validation, output analysis and alternative simulation methods (discrete-event, system dynamics and agent based). Professor Robinson is co-founder of the Journal of Simulation and President of the Operational Research Society. Home page: www.stewartrobinson.co.uk.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

TOWARDS A METHODOLOGY FOR BUILDING LARGE-SCALE DISTRIBUTED HYBRID AGENT-BASED AND DISCRETE-EVENT SIMULATIONS: THE CASE OF EMERGENCY MEDICAL SERVICES

Ms. Anastasia Anagnostou
Dr. Simon J E Taylor

Department of Information Systems and Computing
Brunel University
Uxbridge, Middlesex, UB8 3PH
anastasia.anagnostou01@brunel.ac.uk, simon.taylor@brunel.ac.uk

ABSTRACT

How do we build large-scale models? Can large-scale models be built from models from different simulation paradigms? Can existing simulation modelling methodologies guide the development of large-scale models? Distributed simulation (DS) offers the possibility for large-scale simulations to be developed that can be composed of models based on different world views and different simulation packages. Based on experiences in developing a large-scale Emergency Medical Service simulation that consists of distributed agent-based and discrete-event simulations, this paper proposes an emerging methodology for building large-scale distributed hybrid simulations.

Keywords: Simulation Methodology, Distributed Simulation, Large-Scale Modelling, Hybrid Modelling, Agent-Based Modelling and Simulation, Discrete Event Simulation

1 INTRODUCTION

A simulation study requires much effort, time and technical expertise by researchers and practitioners to plan, develop and conduct experiments with computer simulation models. A simulation project is not just about the development of a computer simulation program but rather involves multiple tasks, all of which are equally important to the successful completion of the project. Arguably, having a clearly defined methodology with steps to follow during a simulation project can save unnecessary work and reduce the possibility of errors.

Several examples can be found in literature that have attempted to frame the process of conducting a simulation study. Ulgen et al. (1994) tackled this issue from a practitioner's viewpoint. They analysed the phases of a simulation project as a set of guidelines mainly for supporting future modellers and concluded in an eight-phases methodology with distinguished steps defined for each phase. The 10-step methodology of Law and Kelton (2000) appears as a sequence of distinguished tasks that are performed with some iteration during the conceptual and coding validation processes. Banks et al. (2000) introduced some parallelisation of activities, in addition to the iterative processes at the experimentation phase.

Another approach is the cyclic simulation methodology of Robinson (2004). He describes the key stages in a simulation study and with double-headed arrows that forming a cycle indicates that there is movement between the key stages during the simulation project.

Distributed simulation (DS) offers the possibility for large-scale simulations to be developed that can be composed of models based on different world views and different simulation packages. Are any of the methodologies introduced above usable for this type of simulation? In this paper, building on these methodologies, we investigate and present an emerging methodology for distributed simulation projects. The development of the proposed methodology is based on experiences in developing large-scale simulations when we attempted to build an emergency medical services (EMS) simulation (Anagnostou, Nouman and Taylor, 2013). In this, we used simulation techniques that were appropriate for the system being modeled: agent-based modelling and simulation (ABMS) for the ambulance service and discrete event simulation (DES) for the A&E departments. We also investigated using distributed and non-distributed approaches. In our EMS simulation we used a hybrid approach combining agent-based modelling and simulation (ABMS) and discrete event simulation (DES) technologies in a DS environment. We addressed issues of data and time synchronisation in the distributed simulation, the semantic differences between the two simulation world views, and model reusability issues.

This paper is organised as follows. In the next section, we discuss issues that modellers may facing when developing EMS, hybrid approaches and distributed and non-distributed (conventional) approaches to large-scale modelling. Observations are drawn from these discussions and form the basis for our emerging methodology for constructing hybrid ABMS-DES distributed simulation models. A concluding section summarises the paper.

2 ISSUES ON MODELLING THE EMERGENCY MEDICAL SERVICES

Emergency medical services (EMS) are complex and multidimensional systems that provide immediate care to patients with acute illnesses or serious injuries. In the past, the role of EMS was to offer transport to those patients that were unable to get to hospital by their own means. Today, at least in developed countries, EMS offers pre-hospital care on the site of incident and during transport to the hospital.

EMS are accessible to the public by an emergency telephone number that put them through a control centre. The control centre personnel initially assess the incidents and find and dispatch the appropriate emergency vehicle and crew. EMS can be public, private or voluntary organisations. Usually, the offered services are classified into two categories, the Basic Life Support (BLS) and the Advanced Life Support (ALS). BLS deals with less serious illnesses and injuries and the crew does not have medical training, whereas ALS deals with serious illnesses and injuries, where the flashing blue lights are on, and the crew has medical skills, i.e., paramedics and emergency medical technicians.

The structure and functionality of EMS vary considerably worldwide. However, this project focuses on the UK's EMS and thus from now on when the term "emergency medical services" is mentioned the reference is for the UK's EMS.

As stated earlier, EMS offers pre-hospital care. However, its role does not end there. Another important and critical offered service is to transport patients, unable to travel by their own means, to hospitals. A timely response and transfer to the regional hospitals' accident and emergency (A&E) departments, more than often, has saved the life of the patient. It is clear that there is close collaboration between the ambulance service and A&E departments. As it is evident from the published literature, the majority of EMS simulation projects consider only the ambulance service up to the point of patient handover. In this research, a comprehensive perspective of EMS is taken in order to perceive a global view of the emergency systems. Thus, in order to model the whole EMS, the A&E departments of the hospitals should be included in a realistic simulation model and operate in collaboration with the ambulance service.

3 HYBRID SIMULATION OR SINGLE APPROACH?

As discussed in the previous section, EMS consists of the ambulance service and the A&E departments, usually situated in the hospitals of the coverage area. The fact that there are involved two different organisations raises the question, whether a single simulation technique is capable to accommodate the functionalities of those two organisations.

The ambulance service model, in a rough outline, includes the emergency call centre, or centres, the vehicles and the crews. The call operators have to respond to the emergency call, assess the incident severity in order to send the appropriate vehicle and crew, find the closest available vehicle to the site of the incident and send the vehicle to the patient. The crew, in turn, apart from the medical treatment on site, has to decide whether the patient needs to be transferred to a hospital or released after the on-scene treatment. If the patient has to be transferred, the closest available hospital and the fastest route should be found. All the above indicate a high degree of interaction among the system's objects. Therefore, the most promising simulation technique to realistically capturing and represent all the interactions of an ambulance service model is ABMS. One can argue that an ambulance service system can be modelled using DES, if the focus is on the processes of the system. However, the presented methodology was emerged from a study that the focus was on the interactions of the individual objects of the ambulance service system.

On the other hand, the A&E departments in a hospital are highly process oriented organisations. When a patient arrives at the A&E, generally and regardless of the mean of arrival, a treatment decision is made according to the patient's condition. If the required resources are available (e.g. cubicle, bed, nurse or doctor etc.), the patient progresses through the system at the next activity. Otherwise, the patient enters a queue until the resources become available. Similarly, in DES, an entity passes through a system's processes, being processed when there are available resources or waiting in a queue as appropriate. An entity's state changes in accordance with the model's activities. However, the entity itself is not able to take decisions or interact with the objects of the simulation, but rather it is driven by the system's processes until, eventually, exits the simulation. DES therefore appears to be the most appropriate for A&E. The sane argument about the simulation paradigm can be presented here, however, in the same way as above, the focus of the study was on the processes of the A&E departments.

3.1 ABMS and DES for Emergency Medical Services

DES is a technique that models a system's behaviour as discrete events. That is an instant of time at which an entity enters or leaves an activity. An activity changes the state of an entity (Pidd 2004). For example, if the entity is a customer awaiting some service, while in the queue, the state of the customer could be "awaiting to be served". Once the server is available, the service will start, this event will change the state of the customer to "being served". DES is being used in various industries to analyse process behaviours within a system (Brailsford et al. 2009; Robinson, 2005).

Furthermore, ABMS is a contemporary simulation technique and is characterised by the agents as individuals that have certain properties. The agents interact with other agents and the environment of the system. As a result, these interactions change the agents' properties, which define the agents' behaviour. ABMS is being used mainly to analyse individual behaviours within a system. Arguably, ABMS is continuously gaining in popularity within the simulation community. One of the reasons is its similarities with object oriented paradigm (North and Macal, 2007).

Regardless the fundamentally different philosophies of the two simulation paradigms, there are significant similarities, too. For example, in both DES and ABMS, the simulation time is progressing in discrete time steps (Pawlaszczyk and Strassburger, 2009) and their modelling view is a bottom-up approach of a system's behaviour.

3.1.1 Semantic relationships between ABMS and DES

To achieve interoperability between systems modelled with different simulation techniques, the semantic relationships between the fundamental notions of these techniques should be defined.

The presented EMS framework involves interoperability between ABMS and DES models. Both paradigms can be described as discrete time-stepped simulation techniques. As opposed to continuous time models, the simulation progresses by advancing from one event to the next. However, ABMS is a time-driven systems, that is, the simulation progresses according to a pre-set time step, while DES is an event-driven systems, that is, the simulation progresses according to a schedule of events. In the latter case, if there is no event scheduled for the next simulation time unit, the software will "jump" to the simulation time unit that an event is scheduled to happen.

Nonetheless, the individual objects in ABMS are active objects that possess behavioural rules, while in DES are passive objects that are driven by the processes in the simulation. In ABMS *resources*, a fundamental component of DES, are *agents*. In DES, there is a distinct meaning of a "*queue*" (has an order, hold entities, etc.) while in ABMS, the agents just *wait* for an activity to begin (not in a queue). *Property* and *attribute* hold the same underlying meaning in ABMS and DES simulation, respectively. *Event* and *activity* are two terms that are met in both techniques and share the same meaning. In ABMS, a set of *rules* defines the steps that the agent will follow, and can be associated with the *system-level rules* of DES that define the process that an entity will flow through.

The *environment* in DES serves visualisation purposes, only, and the entities are typically not aware of it. In ABMS, the agents interact with the environment and learn from it. Finally, the term *state* holds similar semasiology in both techniques. All the above are summarised in Table 1 where the semantic relationships of the basic notions between the two simulation paradigms are listed.

Table 1 *Semantic relationships between ABMS and DES*

ABMS		DES	
Term	Semasiology	Term	Semasiology
Agent	Active object that make decisions and change its behaviour.	Entity Resource	Passive object that flows through the processes.
Waiting	Agents just waiting to perform an activity.	Queue	A building block that stores the entities while waiting for a service. It agrees with the queuing theory.
Property	Defines the agent's characteristics.	Attribute	Defines the entity's characteristics.
Event	A specific time that an activity begins or ends.	Event	A specific time that an activity begins or ends.
Activity	Action that starts and ends with events.	Activity	Action that starts and ends with events.
Rule	Defines the routine that an agent will follow.	System-level rules	Defines the process that an entity will flow through.
Environment	Agents can interact with the environment and learn from it.	Environment	Entities are not aware of it.
State	The set of global variables and all agent's properties at a specific time. The state of an agent can change by an activity.	State	The set of global variables and all entity's attributes at a specific time. The state of an entity can change by an activity.
Time-driven	The simulation progresses according to a pre-set time step.	Event-driven	The simulation progresses according to a schedule of events.

4 DISTRIBUTED VERSUS NON-DISTRIBUTED SIMULATION

Large-scale simulation models that contain more than one organisational subsystem can be modelled either as a single simulation or as separate, independent models that are linked in some form of distributed system. There are various motivations for using distributed simulation against attempting to reuse models by composing them and using a single simulation package. Explored in full in Taylor et al. (2012), these include:

Data transfer/access problems A simulation often draws data from local data sources. As soon as a simulation leaves its domain of use the data must go with it. Data sources can be very large, multiple and/or connected to real-time sources. Moving these and ensuring that they are up to date may not be particularly convenient. This can lead to inaccuracies in results due to inconsistent data.

Privacy and data sharing issues Privacy issues may prevent model sharing across different organisations or even across different departments in the same organization.

Model composability issues Composing several models together in the same simulation software is not a simple 'cut and pasted' procedure, even if the models have been initially developed using the same simulation package. Variable name clashes, global variables and different validation assumptions are three examples of the many problems of this approach.

Model execution time As models grow in size, their process demands may increase predominately due to large event lists. Storage (RAM in particular) may also be strained by the larger model's demands.

It is attractive to think that the creation of large, distributed models that implement local changes efficiently and share the processing load of the model across several computers is possible. However, DS itself is not without its problems and can be extremely complex and often difficult to implement. Therefore, a well-defined simulation methodology for developing such simulation projects can be extremely useful and prevent inconsistencies in the model building process. In the next section, the development of the simulation methodology presented in this paper is explained.

5 DEVELOPING A METHODOLOGY FOR LARGE-SCALE HYBRID ABMS-DES DISTRIBUTED SIMULATIONS

In this section the background and rational of the proposed simulation methodology is stated. Moreover, the development process is explained.

5.1 Background and rational

In the field of modelling and simulation, a lot of discussions have been made about formalising the area as scientific discipline. A common outcome of these discussions is that modelling and simulation needs standard terminology and procedures. More specifically, in the area of distributed simulation, that can be considered as a specialty of modelling and simulation, there is a further requirement. That is, to define the interoperability level among the participating models.

Tolk and Muguira (2003) introduced the idea of a layered view of the stages in distributed simulation projects. Inspired by systems engineering and the Levels of Information Systems Interoperability (LISI) model, Tolk and Muguira established the Levels of Conceptual Interoperability Model (LCIM) as a framework for simulation models composability. The current form of LCIM is shown in Figure 1.

Wang, Tolk and Wang (2009), discuss the underlying notions of the model. Seven levels are defined in LCIM. Ranging from level zero (L0), where there is no interoperability at all, to level six (L6), where conceptual interoperability is achieved. Level one (L1) refers to the technical interoperability and describes the communication protocol between the interoperating models. Level two (L2) refers to syntactic interoperability. That is, the common data structures of the shared variables. Directly higher in the hierarchy stands level three (L3), the semantic interoperability, where there is a semantic mapping of the terms that are used in the interoperating models. For example, in ABMS and DES interoperating models, agents and entities, respectively, refer to the same object of the distributed simulation, namely the object that its state is changed when certain events are happening. Level four (L4) is the pragmatic interoperability, that is the common workflow of the

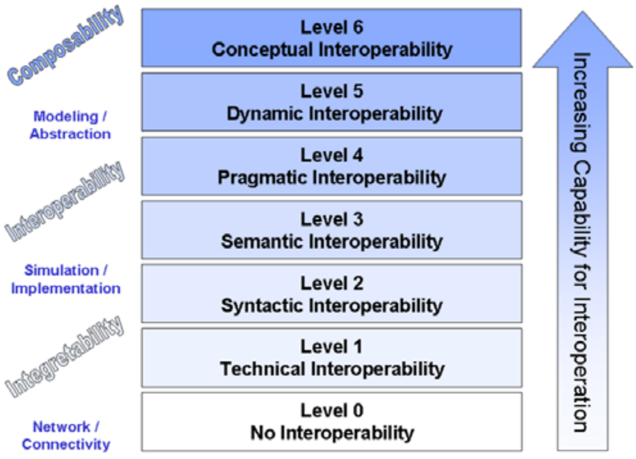


Figure 1 *Levels of Conceptual Interoperability Model (source: Wang, Tolk and Wang 2009)*

interoperating models and defines the context of the exchanged information. The next level five (L5) refers to the dynamic interoperability of the distributed model. Here, the effect of the flow of information during execution is defined. Finally, the highest level in LCIM is level six (L6), where conceptual interoperability is achieved.

The above levels are categorised in three distinctive layers (Page, Briggs and Tufarolo, 2004). The higher levels are included in the composability layer. Composability belongs to the model territory and is the objective of the lower layers. The immediate lower layer is the interoperability layer and describes the implementation issues, such as software details, data exchange, etc. Lastly, the lowest layer is the integratability category and deals with the physical connections of the distributed simulation.

Composability is the highest level in the LCIM. To achieve composability all the underlying levels must be achieved. As very clearly stated by Petty and Weisel (2003), interoperable models are not necessary composable. Interoperability is necessary but not sufficient to achieve composability. By achieving composability, the component models of a DS can be recombined with each other or with different interoperable models and form different DS systems. Therefore, simulation model reusability is supported by composable models.

Another distinction of composability is made by Paul and Anderson (2003). They differentiate the composition of models "of same resolution" and "of different resolution", with the term resolution, here, to suggest the abstraction level of the models. The former is described as horizontal composability and involves composing models of the same level of abstraction. For example, horizontal composability is when all the interoperating simulations are modelled in the individual entity level, namely microscopic simulations. The latter is described as vertical composability and involves models with different abstraction levels. For example, vertical composability is when some of the component simulations are modelled in the individual entity level and some others in the organisation level, namely a combination microscopic and macroscopic simulations. The horizontal composability is relatively easier to be achieved than the vertical composability. However, as the number of horizontal objects, i.e., entities, of the models increases so does the complexity of the distributed system. Furthermore, difficulties lay in the realm of semantics that the different domains the component models may belong.

5.2 Distributed simulation methodology development process

As derived from the background analysis in the previous subsection, a simulation project can be conceptualised in a layered structure. The proposed simulation methodology can be considered as a guideline framework for developing large-scale distributed ABMS-DES simulation models and is based on the simulation methodology that was formed by Banks et al. (2000).

The proposed framework was developed in a three-part approach. In Figure 2, the complete process of the framework development is depicted. The first part of the process is a bottom-up approach and involves the aggregation of the simulation steps into phases. The second part of the distributed simulation methodology development process is a horizontal approach and involves the mapping of the standalone simulation project phases with the distributed simulation project phases. Finally, the third part is a top-down approach and involves the disaggregation of the distributed simulation project phases into distinctive steps. The three parts are analysed in the following subsections.

5.2.1 Development part one

The first approach taken is a bottom-up view of the steps in a standalone simulation project. The slightly altered stepped-process of Banks et al. (2000) simulation methodology is shown in Figure 3.

The first step of the process involves the problem formulation. In this step, a modeller should state clearly the real-world problem that the particular simulation project will attempt to solve or analyse.

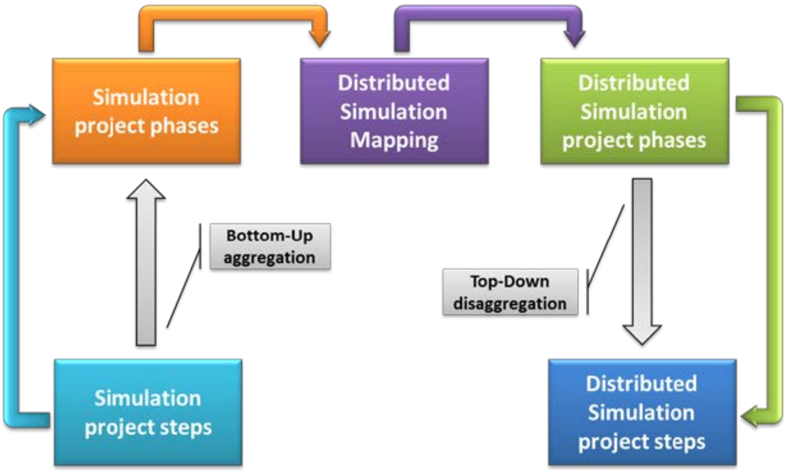


Figure 2 *Framework development approach*

The next step is a parallel activity of starting the collection of data and conceptualising the model, as well as, the selection of the simulation software or language to be used for the realisation phase. Sequentially, but in parallel with data collection, the model realisation phase starts. The model realisation is actually the coding of the simulation software program. Model verification refers to debugging the computer simulation program. Debugging is an iterative process and is happening throughout the coding process. The parallel processes of data collection and model realisation should finish together in order to populate the model with the actual data and be able to start the validation process. The simulation program is valid when it does what was intended to do by its design. Usually the validation process involves pilot runs and testing the results against some performance measures of the real-world system. If the model is not valid, the conceptual model and the available data should be revisited. When the model is valid, the process of designing the experiments can commence.

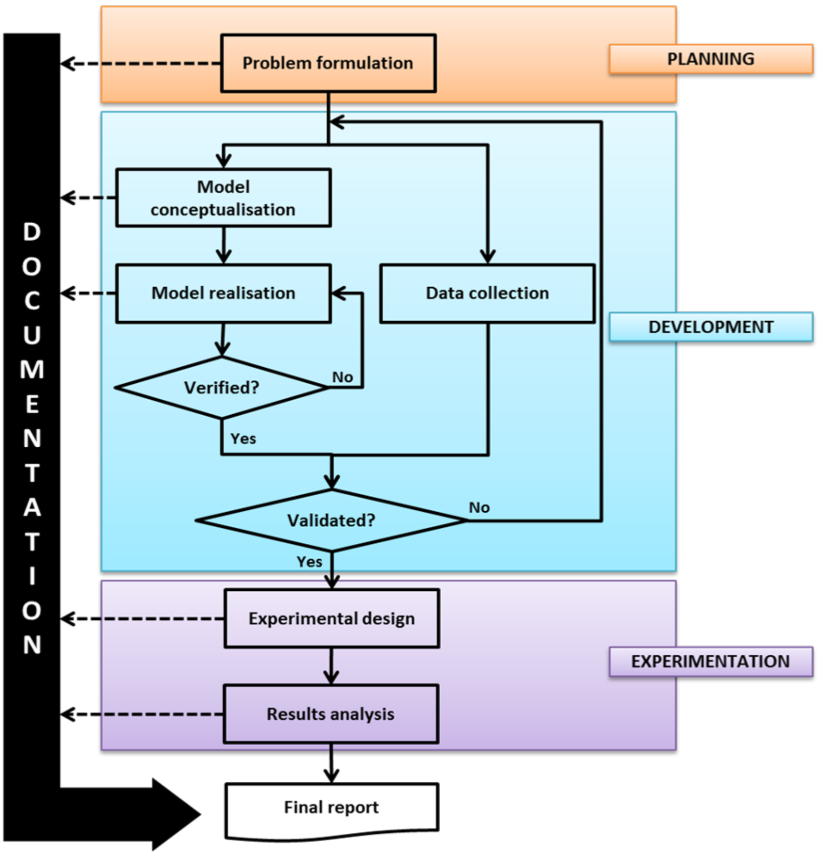


Figure 3 *Simulation methodology for a standalone simulation project*

Finally, the experiments can be conducted and the produced results can be analysed.

From the first step, the documentation process commences and continues throughout the simulation project. This documentation leads to the final report of the project. The small alterations of Banks et al. (2000) framework are the two parallel activities. Namely, the data collection and the documentation processes. From experience, the computer simulation program can be verified even when the model is not populated with the actual data. The model coding, as well as the data collection, process can be very time-consuming. By performing the two activities in parallel fashion, valuable time can be saved. Furthermore, as mentioned in Onggo and Hill (2014), the two activities are closely interwoven. For example, the data availability, or rather unavailability, and quality can lead to model alterations. Similarly, the parallel documentation of each step can help to avoid omissions in the final report.

As mentioned earlier, the conceptualisation of a simulation project can be seen in a layered layout. Therefore, the bottom-up approach that commences the proposed methodology involves the aggregation of the aforementioned steps into layers, or phases as mentioned in this study. The complete process of building a simulation study can be divided into three phases. That is, the planning phase stands in the higher level and concerns the pre-modelling activities, the development phase forms the middle level and concerns the actual modelling activities, and the experimentation phase is the lowest level and concerns the post-modelling activities.

5.2.2 Development part two

The second part in the proposed methodology development rationale is a horizontal approach. This part involves the mapping of the processes of the standalone simulation project with the DS project.

The first phase of a DS project is similar to any single simulation project. That is, the problem formulation, where the objectives of the study should be clarified. However, careful considerations should be made in the development and experimentation phases (see Figure 4).

This part represents an abstract relationship between a standalone simulation project and a distributed simulation project. During this part, a modeller can decide whether a system under study is better accommodated by DS or a single simulation model and whether any existing subsystems models can be reused a part of a distributed system.

In the development phase, the conceptualisation and realisation procedures exist. The conceptual design for the distributed system, mainly, involves the interactions between the participating simulation models. At this point the individual models can be considered as highly abstract entities (or black boxes) and the objective is to identify the boundaries between the interacting models. In other words, the first, and very important, activity is to define the Interoperability Reference Models (IRM). Details in IRMs standardisation can be found in Taylor et al. (2012).

The second procedure in the development phase is the realisation process, and that involves the simulation software development. The individual participating models either exist already or have to

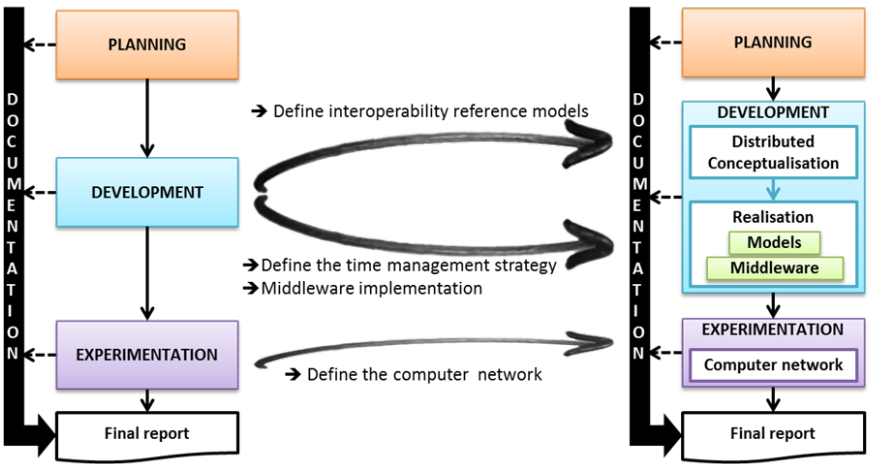


Figure 4 *Mapping the phases in a standalone and a distributed simulation project*

be built from scratch. In the former case, the models should be modified. In the latter case, the models will be built according to the DS conceptual design. The middleware implementation will be responsible for the data and time synchronisation.

In a non-distributed simulation, the only option is to execute the experiments in a single computer. Conversely, DS systems, as the name infers, run in different machines. These can be processors connected via a local network, the internet, or the cloud, or even different cores in the same processor.

5.2.3 Development part three

The third part of the development of the proposed methodology involves the disaggregation of the DS project phases into detailed steps (see Figure 5). For this part, a top-down approach is adapted as mentioned earlier.

The more important differences between a non-DS and a DS project lay in the development layer of the building process. The first activity in the development phase is to conceptualise the distributed system. During this process, the interactions between the simulation models should be defined. This involves the development of the interoperability reference model of the distributed system. The IRM defines the exchanged information but does not necessarily include time synchronisation information. The time dimension is only mentioned when there may be a conflict and the order of events must be in the specified relationship. Further, in the conceptualisation stage, the semantic relationships between the interacting models should be clarified. If the system is homogeneous, i.e., all participating models are modelled with the same simulation paradigm, this is not necessary. However, if the system is hybrid, as is in this paper, there should be a semantic correlation between the different simulation paradigms. At the point, the software tools should be decided.

Then, the next activity in the same phase is the model realisation stage. The realisation stage includes, first, the participating models building and, second, the middleware implementation. At the first part, there are three possible scenarios: a) none of the individual simulation models exist, b) all of the individual simulation models exist, or c) some of the individual simulation models exist. In case (a), the models should be built from the beginning, following the usual procedure. However, when conceptualising the component model, certain considerations should be taken into account for the sake of interoperability. This includes, the transparency level of each component model (what information each model allows to be visible), the global variables of the distributed system and the ownership of these variables (which model can update the global variables). In case (b) the local models already exist, therefore, they should be modified in order to become interoperable. Lastly, in case (c), there will be a combination of both the above procedures.

The second part of the realisation stage includes the DS middleware. The main role of the middleware is the synchronisation of data and logical time management. The synchronisation scheme can be either conservative or optimistic. A conservative scheme does not allow processing of local events in a future logical time of the system, while in the optimistic approach there can be violation of this rule but there is a rollback mechanism to recover from errors and process always the event with the lower timestamp. Furthermore, the logical time advance service should be decided before commencing the actual coding of the middleware interface. For example, in the High Level Architecture (HLA) standards for DS that is used in the present project, there are two such services, namely, time advance request (TAR) and next event request (NER). Each model asks the HLA Run-Time Infrastructure (RTI) for advancing the logical time. The RTI will then send all the messages (i.e., attribute updates etc.) to the interested models and then grants the advance of the logical time (Fujimoto and Weatherly, 1996). Usually, time-driven simulations are facilitated by TAR while event-driven simulations are facilitated by NER.

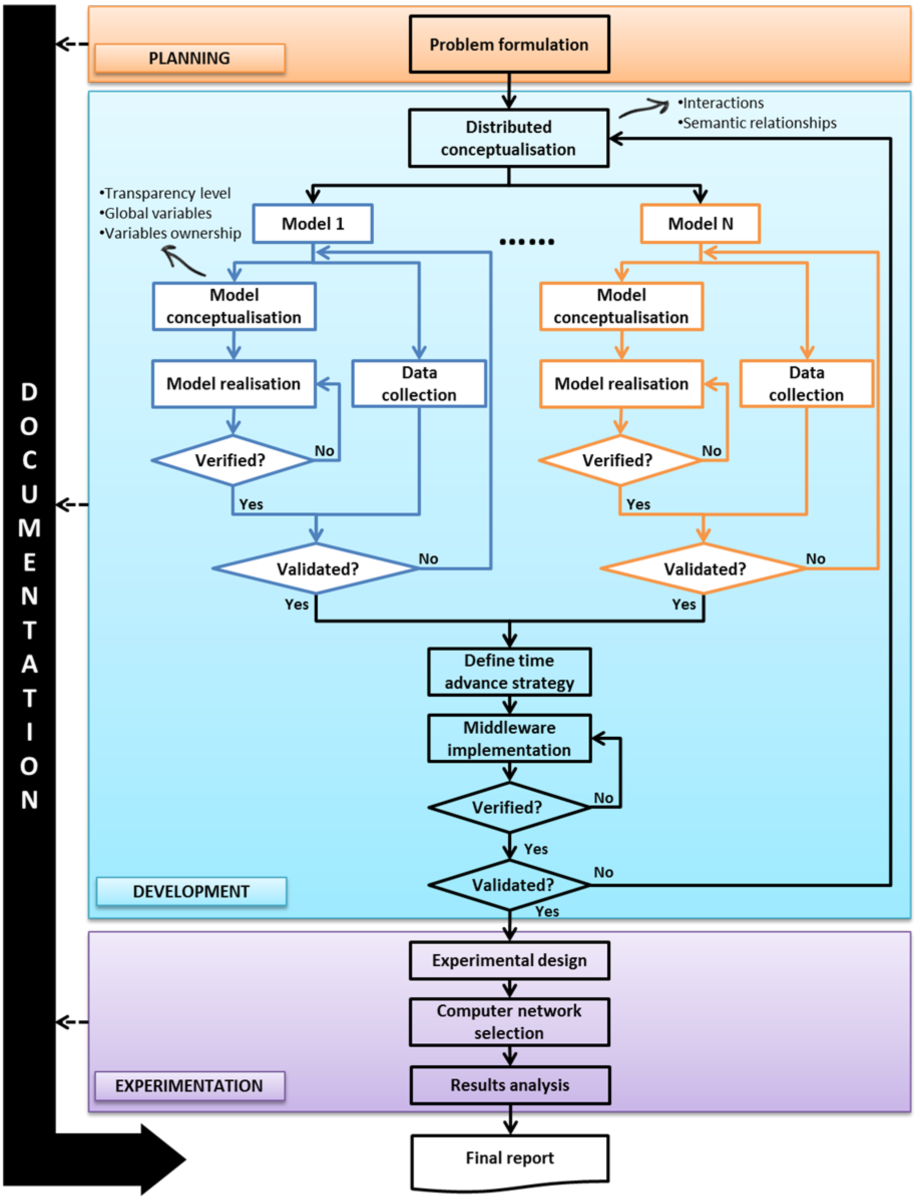


Figure 5 *Distributed simulation methodology*

Once the time advance service is defined, the actual coding of the middleware can begin. Debugging of the software is happening iteratively until the middleware is verified. Afterwards, the complete distributed system should be validated against specified performance measures. If the system is not valid, the distributed model should be checked, starting from the distributed conceptualisation step.

The next layer that differs between the non-DS and the DS projects is the experimental design. As mentioned above, the computer network for the DS projects' experiments execution should be defined, i.e., private network, public network.

6 CONCLUSIONS

In this paper, the development process of a simulation methodology for distributed hybrid ABMS-DES simulation projects was presented. The case of EMS was used as a system case study for explaining the rationale behind the need for such a simulation technology. The proposed simulation methodology was developed based on experience in conducting hybrid ABMS-DES distributed simulation studies on emergency medical services. It can be used as a guide for large-scale simulation projects that involve multi-paradigm distributed simulations. Further research needs to be conducted in order to include more simulation techniques, as well as, different application sectors other than EMS. In so doing, the presented distributed simulation methodology can be extended and generalised, and consequently constitute a general framework for distributed simulation projects that concern about model reusability.

ACKNOWLEDGMENTS

This research is funded by the Multidisciplinary Assessment of Technology Centre for Healthcare (MATCH), an Innovative Manufacturing Research Centre (IMRC) funded by the Engineering and Physical Sciences Research Council (EPSRC) (Ref: EP/F063822/1).

REFERENCES

Anagnostou A, Nouman A and Taylor S J E (2013). Distributed Hybrid Agent-Based Discrete Event Emergency Medical Services Simulation. In: Pasupathy R, Kim S H, Tolk A, Hill R and Kuhl M E (eds.) *Proceedings of the 2013 Winter Simulation Conference*. Washington, DC.

Banks J, Carson J S, Nelson B L and Nicol D M (2000). *Discrete-Event System Simulation*. Prentice Hall Inc.: USA.

Brailsford S, Bolt T, Connell C, Klein J H and Patel B. (2009). Stakeholder Engagement in Health Care Simulation. In: Rossetti M D, Hill R R, Johansson B, Dunkin A and Ingalls R G (eds). *Proceedings of the 2009 Winter Simulation Conference*. Austin, Texas, pp 1840-1849.

Fujimoto R and Weatherly R (1996). Time Management in the DoD High Level Architecture. In: *Proceedings of the 1996 Workshop Parallel and Distributed Simulation*. IEEE CS Press: Los Alamitos, CA, pp 60-67.

Law A M and Kelton W D (2000). *Simulation Modeling and Analysis*. 3rd Edition, McGraw-Hill Book Co: Singapore.

North M J and Macal C M (2007). *Managing Business Complexity*. Oxford University Press Inc.: New York.

Page E H, Briggs R, Tufarolo J A (2004). Toward a family of maturity models for the simulation interconnection problem. In: *Proceedings of the Spring 2004 Simulation Interoperability Workshop*. IEEE CS Press: Arlington, VA.

Pawlaszczyk D and Strassburger S (2009). Scalability in distributed simulations of agent-based models. In: Rossetti M D, Hill R R, Johansson B, Dunkin A and Ingalls R G (eds). *Proceedings of the 2009 Winter Simulation Conference*. Austin, Texas, pp 1189–1200.

Petty M D and Weisel E W (2003). A Composability Lexicon. In: *Proceedings of the IEEE Spring Simulation Interoperability Workshop*. IEEE CS Press: Kissimmee, Florida.

Pidd M (2004). *Computer Simulation in Management Science*. 5th Edition, John Wiley & Sons Ltd: Chichester, UK,

Davis P K and Anderson R H (2003). *Improving the Composability of Department of Defense Models and Simulations*. RAND Corporation: online (last accessed 12 Nov 2013) http://www.rand.org/pubs/monographs/MG101.html

Onggo, B S S and Hill J (2014). Data Identification and Data Collection Methods in Simulation: A Case Study at the ORH Ltd. *Journal of Simulation* advance online publication 17 January doi:10.1057/jos.2013.28.

Robinson S (2005). Discrete-Event Simulation: From the Pioneers to the Present, What Next? *Journal of the Operational Research Society* **56(6)**: 619-629.

Robinson S (2004). *Simulation – The Practice of Model Development and Use*. John Wiley & Sons Ltd: Chichester, UK.

Taylor S J E, Turner S J, Strassburger S and Mustaffee N (2012). Bridging The Gap: A Standards-Based Approach to OR/MS Distributed Simulation. *ACM Transactions on Modeling and Computer Simulation* **22(4)**: article 18.

Tolk A and Muguira J A (2003). The Levels of Conceptual Interoperability Model. In: *Proceedings of the IEEE Fall Simulation Interoperability Workshop*. IEEE CS Press: Orlando, Florida.

Ulgen O M, Black J J, Johnsonbaugh B and Klungle R (1994). Simulation Methodology – A Practitioner's Perspective. *International Journal of Industrial Engineering, Application and Practice* **1(2)**: online (last accessed 12 Nov 2013) http://pmcorp.se/PublishedPapers/Simulation%20Publications/SimMethodology/SimulationMethodology_APractitionersPerspective.pdf

Wang W G, Tolk A, Wang W P (2009). The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to Modeling and Simulation. In: *Proceedings of the Spring Simulation Multiconference.* Society for Modeling & Simulation International (SCS): San Diego, CA.

AUTHOR BIOGRAPHIES

ANASTASIA ANAGNOSTOU is a PhD candidate at the School of Information Systems, Computing and Mathematics, Brunel University, UK. She holds an MSc in Telemedicine and e-Health Systems and a BSc in Electronic Engineering. Her research interests are related to the application of modelling and simulation techniques within the healthcare sector. Her email address is Anasta-sia.Anagnostou01@brunel.ac.uk.

SIMON J.E. TAYLOR is the convener of this new phase of Grand Challenge activities. He is the Founder and Chair of the COTS Simulation Package Interoperability Standards Group under SISO. He is the Editor-in-Chief of the UK Operational Research Society's (ORS) Journal of Simulation and the Simulation Workshop series. He was Chair of ACM's SIGSIM (2005-2008) and is a member of the SIGSIM Steering Committee. He is a Reader in the Department of Information Systems and Computing and Brunel and leads the ICT Innovation Group. He has published over 150 articles in modelling and simulation. His recent work has focused on the development of standards for distributed simulation and grid- and cloud-based simulation in industry (www.cloudsme.eu) as well as the spread of the international Grid Infrastructure into Africa (www.eI4Africa.eu). His email address is simon.taylor@brunel.ac.uk and his web address is www.brunel.ac.uk/~csstsjt.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

TOWARDS AUTOMATED SIMULATION INPUT DATA

Panagiotis Barlas

Enterprise Research Centre
University of Limerick
Limerick, Ireland
panagiotis.barlas@ul.ie

Cathal Heavey

Enterprise Research Centre
University of Limerick
Limerick, Ireland
cathal.heavey@ul.ie

ABSTRACT

Discrete Event Simulation (DES) has proved itself to be an effective tool for complex processes analysis. The drawback of using DES is the effort required and costs spent on collecting and processing the input data from different data sources. To address the problem of time consuming pre-coding for DES projects, a tool based on Core Manufacturing Simulation Data (CMSD) is currently being developed. The tool will read data from several resources of an organisation; analyse it using statistical analysis and output it in a format that is applicable to simulation purposes. The format that we adopted follows the CMSD standard in order to describe simulation related data. The CMSD specification is presented as simulation input, in order to achieve the efficient reuse of this data for future DES projects. We present a first prototype and a test implementation of this tool and we draw conclusions about the future steps of our project.

Keywords: Discrete Event Simulation, CMSD, RPy2

1 INTRODUCTION

Manufacturing organisations, procedures, and data are growing and becoming more complicated. Manufacturing engineering, product design, production and industrial management decisions include the consideration of many co-dependent factors and variables, which create many complexities in decision making. Discrete Event Simulation (DES) is one of the most effective tools for planning, designing and improving material flows in production. DES can be used to study and compare alternative designs or to solve problems on existing systems (Fowler 2004).

The widespread use of simulation technology is hindered in the manufacturing industry due to a number of technical and economic barriers. The cost of applying and using simulation technology is high. The cost of integrating simulation systems with other manufacturing applications is even higher. The cost of implementing DES is excessive owing to the cost of Commercial-Off-The-Shelf (COTS) DES packages and the time needed for companies to spent on learning how to use them. The users of commercial DES packages deal with problems when they want to customise the standard provided objects or even develop completely new ones (King and Harrison, 2010). For this reason organisations prefer to have specific, tailored to their own needs simulation based solutions.

Large companies continuously record raw data, and are therefore able to collect large quantities of resource event information. However, usually it is difficult to extract and reuse data for future DES projects. The difficulty is based on the sharing between data files and simulation models. There is always a critical need to extract, analyse and input data from an organisation’s Enterprise Resource Planning (ERP) system, Manufacturing Execution Systems (MES), Material Requirement Planning (MRP) systems or other data sources to the simulation model. The development of a tool that covers the above needs and the use of reusable, neutral standard interfaces would facilitate to reduce the costs related to simulation model development and enable data exchange between simulation and other manufacturing applications. That would make simulation technology more friendly and reachable to a larger range of industrial users and particularly Small and Medium Enterprises (SMEs).

"Decision support in Real-time for Efficient Agile Manufacturing" (DREAM) is an FP7 project, which started in October of 2012. The scope of DREAM (http://dream-simulation.eu accessed 6 February 2014) is to increase the competitiveness of the European manufacturing sector through targeting the advancement of DES technology beyond the current state of the art. DREAM will produce an open-source simulation based platform, which shall provide the capability of developing tailored simulation based solutions for SMEs and other organisations.

The DREAM platform will be based on simulation giving a new open-source, expandable and semantic free Simulation Engine (SE). Our work involved progressing the SimPy code (http://simpy.readthedocs.org/en/latest/ accessed 6 February 2014) developing above it a layer of manufacturing objects. We named this layer ManPy standing for "Manufacturing in Python" (Dagkakis et al. 2013). ManPy will co-operate with other modules such as Graphical User Interface (GUI), knowledge extraction tools, such as the tool described here, and optimisation. All the modules should be independent but facilitate collaboration.

In this paper we present a methodology to automate simulation input data. Our approach is to demonstrate a tool for data extraction, analysis and output the outcome of the analysis using the CMSD format. The remainder of the paper is constructed as follows: in the next section we present the CMSD standard, describing the CMSD information model and the research related to this specification. Then we introduce the tool and present a methodology for automated simulation input data. A case study with the implementation of the tool is presented in the fourth section. We end the paper with conclusions and future work to be carried out to achieve the automation of input data into a simulation model.

2 CORE MANUFACTURING SIMULATION DATA

As it is stated in the Introduction it is difficult to extract and reuse data for future DES projects, this difficulty is based around the sharing of information between data files and simulation models. Most of the time, the needed information can be found in various Information Technology systems (IT-systems). However, data is usually not in the right format required for DES and IT-systems do not have a standardized way of communicating with each other. A reusable, neutral, standardized interface should help reduce the effort and cost related to the input of data management in DES projects (Johansson et al. 2007).

Researchers at the National Institute of Standards and Technology (NIST) in collaboration with industrial partners have developed the Core Manufacturing Simulation Data (CMSD) Product Development Group (PDG) under the guidelines and procedures of the Simulation Interoperability Standards Organization (SISO). The idea of the CMSD effort is to facilitate data exchange and sharing by using neutral, reusable data structures for managing actual production operations and for simulating the performance of the manufacturing systems.

According to "Standard for: Core Manufacturing Simulation Data – UML Model" (SISO, 2010) published in September 2010, the purpose of this standard include:

- enabling data exchange between simulation applications and other software applications,
- supporting the construction of manufacturing simulators,
- supporting the testing and evaluation of manufacturing software,
- enabling greater manufacturing software application interoperability.

2.1 CMSD Information Model

The CMSD information model is a standard representation for core manufacturing simulation data. It describes the essential entities in the manufacturing field and the relationship between those entities that are needed to create manufacturing simulations.

The CMSD information model is given in two different modeling languages: (1) the information model defined using the Unified Modeling Language (UML) and (2) the information model defined using the eXtensible Modeling Language (XML) (Leong et al. 2006). The exchange of data between simulations will be enabled through the exchange of XML instance documents that follow the CMSD

XML Schemas. The UML diagram and XML schemas are intended to be equivalent representations of the same CMSD model.

The CMSD model is designed as a suite of interconnected information modeled as UML classes contained within UML packages, presented as a series of UML class and package diagrams (Riddick and Lee, 2008). The manufacturing concepts within each package are modeled as UML classes and the characteristics related to each entity are modeled as UML class attributes. Operation specifications are not used in defining classes and the availability of all class attributes is public (SISO, 2010).

2.2 Research related to CMSD

An Application Programming Interface (API) was created for abstracting away the implementation details of reading and writing CMSD files to/from computer memory (Fournier 2011). Several translators were created to test the applicability of the CMSD effort and the implementation of the CMSD API in different simulation software like Arena, QUEST, FlexSim and others. FACTS is a research project that finished in March 2008 (Ng et al. 2007). The project purpose was to design and evaluate methods and IT-tools for efficient development of new and modified factories. The project resulted in two acknowledged tools; one of them is the GDM-Tool (Skoogh et al. 2012). The GDM-Tool aim is to allow the automatic integration of production data into the DES models stored in different IT-systems at companies (Bengtsson et al. 2009).

In the Simulation-based Manufacturing Interoperability Standards and Testing (SBIT) project was carried out at the National Institute of Standards and Technology (NIST), an aspect of this work is to simulate the production line of an automotive company (Kibira and McLean, 2007). They used the CMSD and they focus on the development of a data driven simulation model of the automotive manufacturing plant. Another approach of model generation using the CMSD information model was illustrated by Bergmann et al. (2011). They developed a so called CMSD based model generator. The model generator itself serves as a transformation layer between CMSD and the used simulation environment. Other work came from the collaboration of researchers at Chalmers University and NIST (Lee et al. 2011), they developed generic and reusable interfaces for CMSD-file communication in two commercial simulation packages, which are Plant Simulation and Enterprise Dynamics.

3 METHODOLOGY FOR INPUT SIMULATION DATA – INTRODUCTION OF THE TOOL

A widely quoted rule in the simulation literature is the “40-20-40” rule, which states that 40% of the time in a simulation project is spent at the “pre-coding stage”, 20% coding and 40% on the analysis phase. To address the issue of efficient implementation of simulation technology in manufacturing companies, research and development is required at the “pre-coding” phase. In our effort we address this issue by advancing and developing further methods for modelling and capturing systems knowledge. In this direction, we are building a tool that facilitates the extraction of required simulation data, the analysis of this data and finally the output in a format that is readable by simulation software.

The tool is aimed to link production data stored in different IT-systems at companies with the simulation software. The philosophy of our project is to use and expand explicitly open-source tools. For this reason, a review of open-source data mining tools, statistical computing packages and business intelligence suites was conducted. Searching both the World Wide Web and academic publications revealed 55 open-source tools. We evaluated these tools under certain criteria on four different groups; these groups were General characteristics, Project activity, Operational characteristics and Data Mining characteristics. From this evaluation we selected the R project. R (http://www.r-project.org/ accessed 6 February 2014) is a free software suite for statistical computing, graphics and tasks associated with data mining.

Having in mind the above tool and that the Python programming language is the core of the DREAM project, since ManPy is by nature a Python library, we build the first prototype of the tool using RPy2. RPy2 (http://rpy.sourceforge.net/rpy2.html accessed 6 February 2014) is an interface between Python, which is a popular all-purpose scripting language, and R, which is a scripting language mostly popular for data analysis, statistics and graphics. This interface gives us the ability to have full access in R functions from Python script. It is well developed and quite active as a project retaining a mailing list and providing thorough documentation. It can be used under the GNU Lesser General Public License (LGPL) which makes it is feasible to be also used in proprietary project (Fogel, 2005). RPy2 built on Python offers many convenient features for building code such as efficient list processing and flexible type casting (Hetland, 2008). The drawback is that Python as a scripting language is slower than static languages such as C++ or Java (Dawson, 2002).

Data is divided into three categories based on availability and collectability (Robinson and Bhatia 1995). The first data category is already available in company IT resources, for instance ERP, or MES, or other database. The second category data requires effort because it has to be gathered during the simulation project and the third category, which is data neither available nor collectable. In the first category data can be found as raw data, automatically gathered scrap quantity or process time in a station. These type of data samples are extracted in XLS files (see Figure 1) from a company's ERP, or MES, or database. After the initial extraction, some process may be needed to transform the samples into a useful form. For instance, to obtain process time of a station in a production line, the stop time has to be subtracted from the start time. Additionally, after having the actual process time data points, this data should be analysed in order to fit a distribution. These calculations and the distribution fitting will be one part of the tool, "Data process" (Figure 1). The other part is "Output preparation", this tool should provide Extensible Markup Language (XML) files that follow the CMSD standard and can be used as input for the ManPy (Barlas et al. 2013) simulation tool. As it is stated above the tool is being implemented, as shown in Figure 1.

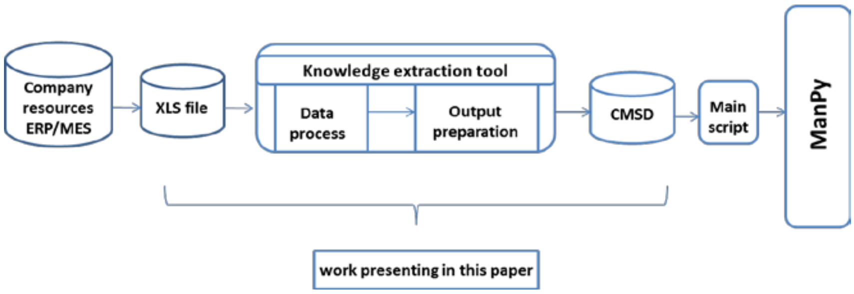


Figure 1 *Data flow diagram in ManPy*

At the time of writing this paper basic statistical functionalities have been developed and linked to CMSD output. It currently can calculate with the use of RPy2 library basic statistical measures like mean value, median, standard deviation, variance, range, interquartile range. Also, it can identify and fit data using the following distribution functions:

- Normal
- Exponential
- Poisson
- Gamma
- Logistic
- Geometric
- Cauchy
- Log-Normal
- Negative binomial

The distribution identification is based on the Kolmogorov-Smirnov test statistics, Calculation of P-value and Maximum Likelihood Estimation, all of which are proven statistical methods. The CMSD information model is implemented in XML with the use of XML.ETREE Python library (http://docs.python.org/2/library/xml.etree.elementtree.html# accessed 6 February 2014), so it can be easily integrated in a script with the output of the above RPy2 functions.

4 TEST IMPLEMENTATION OF THE TOOL

4.1 Description of case study

Several production lines operate in the medical device fabrication facility involved in the DREAM project. The pilot line chosen as a use case for supporting the expansion and validation of the DREAM platform is located in a clean room where other lines also operate; the selection of the pilot line was based on the relatively uncomplicated variations between the different product types. Even though there are dimensional variations in the products fabricated on this line, the pilot line can be considered to be product dedicated, which means the production flow is the same for all products. The product in question is a medical device for use in operating theatres. The line can be considered as an assembly line and different tasks can be assigned to each station for balancing purposes. The process flow consists of four main sections (see Figure 2).

As happens for most of the lines in the plant, the process is labour intensive; each station requires one or more operators. The presence of more than one operator per station is mandatory in the last section where three operators are required to run the two parallel stations. In some sections, the presence of a number of operators greater than the number of stations enables processing time reductions whenever work division is possible.

The medical device industry, realizing the importance of simulation technology, has conducted several simulation based projects in the past. However, none of the simulation models developed could be easily integrated with IT systems used to operate the pilot line to facilitate continuous use of the simulation model. This provided motivation for this company to become involved in the DREAM project.

For input data to the DES model, the medical device company has various manufacturing information, in different formats, stored in databases, ERP system and MES within their organisation. The company in general is data rich and wants to improve the interaction with the stored data. To prepare the input data for the DES model, extraction and processing of the relevant manufacturing information from these databases will be required. This input data process will be both time-consuming and work-intensive. By using the tool to extract and process the data from the organisation IT systems, it is simple to provide updated data with the latest production information in the simulation model.

4.2 The CMSD implementation

In the CMSD standard are offered a variety of classes or categories of manufacturing information which can be used for representing data. In our case we built the CMSD information model of the production line using the most important classes for entering data in a simulation model. The classes that have been used are Resource information, Part information, Process definition and Process Plan information. Resource information describes the people and equipment that perform manufacturing activities. CMSD resource types include machines, stations and employees of the production line. Process plan information specifies the set of production activities needed to transform materials and subcomponents into finished products. Consequently, process plans are intended to specify part routing information. Process definition contains information about the processes in the process plans, information such as indication of each task that will be performed to create a part and resources either machines or employees that are to be used to execute these tasks. The data section in the CMSD file begins with a list of Part types and Resources, followed by Process Plan and Process definitions. The information model for this line is written using XML and it was completed with 723 lines of XML.

In Figure 2 is presented the process flow diagram of the production line. The process flow consists of four sections, for the building of CMSD document these sections are denoted in Figure 2 as PA, PB, PC and PD. The first section PA has two lines (PA1, PA2) operating in parallel in series (see Figure 2), each one of these lines has three stations in series (P1, P2, P3 and P4, P5, P6). Initial processing of the product occurs here before it proceeds in section PB. Section PA stations must operate in sequence on the product. Section PB has just one station and it follows section PC, which consists of two stations (P8, P9) that operate in parallel. The final section of the line contains two stations (P10, P11) that also operate in parallel. Each one of these eleven processes contains information related to processing times, scrap quantity, buffer capacity and operation details.

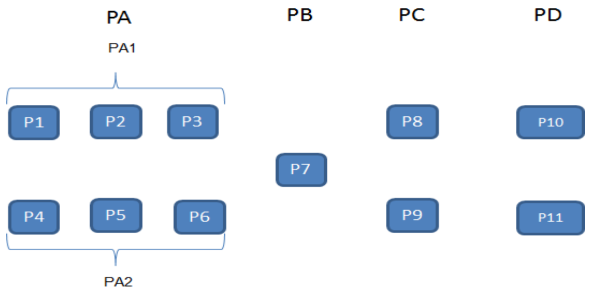


Figure 2 *Process flow diagram of the line*

A process plan object contains one or more process objects (see right part of Figure 3). Each process object may denote either an individual process or a process group. The process plan indicates which process executes first. A process group indicates that a group of processes either executes in a sequence (sequence group), only one process in the group executes (decision group), or all processes in the group execute concurrently. The CMSD standard having these instructions provides flexibility to define most complex manufacturing relationships. In the left part of Figure 3 is depicted part of process plan definition. The XML shows the four different sections (PA, PB, PC, PD) of the line (see Figure 2) and defines that these sections are executed in a sequence. It also defines that the PA section consists of two sub processes that is executed in parallel (decision group).

The right part of Figure 3 shows the process definition in one of the line's stations. They depict the required resources for this operation, first the employee performing the operation and then the resource either station or machine where the operation is being performed. Moreover, the process definition shows the operation time and as a property the scrap quantity of this station; properties used to define characteristics and capabilities of equipment and employees.

```
<ProcessPlan>
    <Identifier>ProcessPlan:PPPlan1</Identifier>
    ....
    <Process>
        <Identifier>PFirst</Identifier>
        <RepetitionCount>1</RepetitionCount>
        <SubProcessGroup>
            <Type>sequence</Type>
            <Process>
                <ProcessIdentifier>PA</ProcessIdentifier>
            </Process>
            <Process>
                <ProcessIdentifier>PB</ProcessIdentifier>
            </Process>
            <Process>
                <ProcessIdentifier>PC</ProcessIdentifier>
            </Process>
            <Process>
                <ProcessIdentifier>PD</ProcessIdentifier>
            </Process>
        </SubProcessGroup>
    </Process>

    <Process>
        <Identifier>PA</Identifier>
        <SubProcessGroup>
            <Type>decision</Type>
            <Process>
                <ProcessIdentifier>PA1</ProcessIdentifier>
            </Process>
            <Process>
                <ProcessIdentifier>PA2</ProcessIdentifier>
            </Process>
        </SubProcessGroup>
    </Process>
```

```
<Process>
    <Identifier>P3</Identifier>
    <Description>StationName</Description>
    ...
    <ResourcesRequired>
        <Description>The employee performing the operation</Description>
        <Resource>
            <ResourceIdentifier>C</ResourceIdentifier>
        </Resource>
    </ResourcesRequired>
    <ResourcesRequired>
        <Description>The resource where the operation is being performed.</Description>
        <Resource>
            <ResourceIdentifier>resource5</ResourceIdentifier>
        </Resource>
    </ResourcesRequired>
    <OperationTime>
        <Unit>minute</Unit>
        <Distribution>
            <Name>Normal</Name>
            <DistributionParameter>
                <Name>mean</Name>
                <Value>4.9849103458</Value>
            </DistributionParameter>
            <DistributionParameter>
                <Name>standardDeviation</Name>
                <Value>1.83374176591</Value>
            </DistributionParameter>
        </Distribution>
    </OperationTime>
    <Property>
        <Name>ScrapQuantity</Name>
        <Value>1</Value>
    </Property>
</Process>
```

Figure 3 *Part of process plan (left) and process definition (right) in XML-based CMSD*

4.3 Implementation of data transformation

In this test implementation, data is extracted into the Microsoft Excel (Excel) format and then imported in the tool and processed with the use of RPy2 library in Python script. After the extraction and the execution of the process, the outcome of the process is mapped onto the CMSD information model, so as to be used as input data into the simulation software.

The tool imports the files with the real data and processes it in order to go from raw data to simulation input information. To do this transformation the following steps are required, all corresponding to Python scripts:

1. Import Ms Excel files.
2. Read the Processing time and the scrap quantity in the stations of the production line.
3. Create lists with attributes the processing time and the scrap quantity for the six stations.
4. Replace missing values in scrap quantity lists with 0 (blank cell in Excel means that was no scrap in this station).
5. Conduct distribution fitting test (Kolmogorov-Smirnov test) for each of the above lists for the processing times.
6. Estimate the rounded mean value for each of the above lists for the scrap quantity.
7. Export the data to a CMSD information model in XML file.

The above steps are all implemented using Python scripts. The tool is built in a way that the data input, the processing of this data and the output preparation are conducted by a separate script. We refer to this script as the “main script”. This main script is the only one to be changed in order to read data from the different Excel files. Therefore, this main script calls different objects so as to give as an output the actual CMSD information model, updated with the new available data. The called objects in our case are described above from step 4 to 7; these are a *ReplaceMissingValues* object, a *DistFitTest* object, a *BasicStatisticalMeasures* object for the rounded mean value and a *CMSD_Output* object.

5 CONCLUSION AND FUTURE WORK

This paper demonstrates a newly developed tool that aims to address the problem of the time consuming pre-coding for DES projects and its tests this implementation using data from an industrial company. The tool is still under development for further enhancement of its capabilities. The current effort includes the ability to import data in different file formats like txt, excel, csv and xml, initial data preparation with the replacement of missing values, calculation of some basic statistical measures, distribution fitting and output preparation again in different file formats but mainly to the use of CMSD, a standard for the exchange simulation based data.

This is preliminary work for the DREAM project and proved the functionality of Knowledge Extraction tool. Our effort targets to achieve or to be as close as possible to the automation of input simulation data. With the term automation we mean that the simulation model automatically collects data from system’s data sources via an interface as and when required to run the model. Next steps in our research are the expansion of the tool adding for example some more distribution functions to fit data, the achievement of the integration of the tool and ManPy under a real use case having as a basis the already integration of CMSD information model with ManPy (Barlas et al., 2013). Another step is to build a user interface for this tool, an interface that should be as friendly as possible to use so as to be easily handled by end users. Finally, validation of the tool through other pilot cases, these cases will provide further opportunity to test, evaluate, improve, and develop the tool, the CMSD specification and generally the integration with ManPy.

ACKNOWLEDGMENTS

The research leading to the results presented in this paper has received funding from the European Union Seventh Framework Programme (FP7-2012-NMP-ICT-FoF) under grant agreement n° 314364.

REFERENCES

Barlas, P., Dagkakis, G. and Heavey, C. (2013). A prototype integration of ManPy with CMSD. In *27th European Simulation and Modelling Conference, EUROSIS 2013*.

Bengtsson, N., Guodong, S., Johansson, B., Lee, Y.T. , Leong S., Skoogh, A. and Mclean, C. (2009). "Input Data Management Methodology for Discrete Event Simulation." In *Simulation Conference (WSC), Proceedings of the 2009 Winter*, 1335–1344.

Bergmann, S, S Stelzer, and Steffen Straßburger (2011). "Initialization of Simulation Models Using CMSD." *Proceedings of the Winter Simulation Conference*, 2223–2234

Dagkakis, G., Heavey, C., Robin, S. and Perrin, J. (2013). ManPy: An Open-Source Layer of DES Manufacturing Objects Implemented in SimPy. In *8th EUROSIM Congress on Modelling and Simulation, EUROSIM2013 (EUROSIM2013).*

Dawson, B. (2002). "Game Scripting in Python." Proc. GDC.

Fogel, K. (2005). Producing Open Source Software: How to Run a Successful Free Software Project.

Fournier, J. (2011). "Model Building with Core Manufacturing Simulation Data." *Proceedings of the Winter Simulation Conference*, 2214–2222.

Fowler, J.W. (2004). Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems. *Simulation*, 80(9), pp.469–476.

Hetland, M. L. (2008). Beginning Python: from novice to professional . 2nd ed. Apress, Berkely, CA.

Johansson, M., Johansson, B., Skoogh, A. , Leong, S., Riddick, F., Lee, Y. T. , Shao, G., and Klingstam, P. (2007). "A Test Implementation of the Core Manufacturing Simulation Data Specification." *Proceedings of the 39th Conference on Winter Simulation: 40 Years! The Best Is yet to Come*, 1673–1681.

Kibira, D. and McLean, C. R. (2007). Generic simulation of automotive assembly for interoperability testing. *2007 Winter Simulation Conference*, 1035–1043.

King, D. & Harrison, H. (2010). Discrete-event simulation in Java: a practitioner's experience. *Proceedings of the 2010 Conference on Grand Challenges in Modeling & Simulation*.

Lee, Y. T., Riddick, F. H. and Johansson B. J. I. (2011). "Core Manufacturing Simulation Data – a Manufacturing Simulation Integration Standard: Overview and Case Studies." *International Journal of Computer Integrated Manufacturing* 24 (8) (August), 689–709.

Leong, S, Lee, Y.T. and Riddick, F. (2006). "A Core Manufacturing Simulation Data Information Model for Manufacturing Applications." *Simulation Interoperability Workshop, Simulation Interoperability and Standards Organization*, 1–7.

Ng, A., Urenda, M. , Svensson, J., Skoogh, A. and Johansson, B. (2007). "FACTS Analyser: An Innovative Tool for Factory Conceptual Design Using Simulation." *Proceedings of the Swedish Production Symposium*, 1–8.

Riddick, F. and Lee, Y.T. (2008). "Representing Layout Information in the CMSD Specification." *Simulation Conference, 2008. WSC 2008. Winter*, 1777–1784.

Robinson, S., and Bhatia, V. (1995). Secrets of Successful Simulation projects. *In WSC'95. 1995 Winter Simulation Conference Proceedings*.

SISO (2010). "Simulation Interoperability Standards Organization (SISO) Standard for : Core Manufacturing Simulation Data ― UML Model." Core Manufacturing Simulation Data Product Development Group, Simulation Interoperability Standards Organization.

Skoogh, A., Johansson, B. and Stahre J. (2012). "Automated Input Data Management: Evaluation of a Concept for Reduced Time Consumption in Discrete Event Simulation." *Simulation 88* (11) (April 23), 1279–1293.

AUTHOR BIOGRAPHIES

PANAGIOTIS BARLAS received his M. Eng in Mechanical Engineering from the Aristotle University of Thessaloniki in 2010. He completed his M. Sc. in Logistics Management at the same University in 2012. He is currently a PhD student in the University of Limerick. His research interests are in Discrete Event Simulation modelling and Data Analytics.

CATHAL HEAVEY is an Associate Professor in the Department of Design & Manufacturing Technology at the University of Limerick. He is an Industrial Engineering graduate of the National University of Ireland (University College Galway) and holds a M. Eng.Sc. and Ph.D. from the same University. He has published in the areas of queuing and simulation modelling. His research interests includes, simulation modelling of discrete-event systems; modelling and analysis of supply chains and manufacturing systems; process modelling; component-based simulation and decision support systems.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

AN INVESTIGATION ON TEST DRIVEN DISCRETE EVENT SIMULATION

Shahriar Asta, Ender Özcan, and Peer-Olaf Siebers

University of Nottingham
School of Computer Science
Nottingham, NG8 1BB, UK
{sba,exo,pos}@cs.nott.ac.uk

ABSTRACT

This paper deals with the application of modern software development tools on simulation development. Recently, Agile Software Development (ASD) methods enjoy an increasing popularity. eXtreme Programming (XP) techniques, one of the techniques which belong to the ASD group of methods is a software development method which improves software quality and responsiveness of software projects through introducing short development cycles and a Test Driven Development (TDD) philosophy throughout the development. In this paper, we particularly pay attention to the application of the TDD by approaching simulation development from a test-first perspective. This study consists of a feasibility study of applying the TDD technique in simulation development in its various levels, say, acceptance and unit testing. Moreover, a simulation case study of a surgical ward has been considered, designed and implemented using the AnyLogic simulation toolkit. Our study differs from the mainstream in many ways. It addresses the feasibility of Test-Driven Simulation Development in Visual Interactive Modelling and Simulation (VIMS) environments as well as providing an insight into how the test-first concept can further help with the choice of components and acceptance testing.

Keywords: Test-Driven Development, Discrete Event Simulation, Extreme Programming, Modelling

1 INTRODUCTION

Agile Software Development (ASD) has been a recent revolution in program design. ASD combines a set of software development methods and is based on iterative and incremental improvement (Beck et al 2001). Many modern software development methods fall into the ASD category, one of which is the eXtreme Programming (XP). XP itself consists of short development cycles where at the end of each cycle the software product is *released* with the goal of absorbing new customer requirements, resulting in an adaptive development cycle. XP is consisted of various elements such as pair programming, various code review methods as well as unit testing. Unit testing in XP is particularly interesting considering that XP intends to promote the software product quality by employing a test-first concept which gives rise to Test Driven Development (TDD) methodology (Beck 2002). TDD requires a programmer to take a different view of the system by writing (and thinking about) the test cases before writing the actual unit code. Instead of being a method to test the software, TDD is the methodology of helping the developer and customer with specifying "*unambiguous requirements*" which are expressed in form of tests (Newkirk 2004). The TDD development cycle is shown in Figure 1. As well as being heavily utilized in industrial scale (Bhat and Nagappan 2006; Larman and Basili 2003; Gelperin and Hetzel 1987), TDD has been the subject of numerous research publications (Newkirk 2004; Erdogmus et al 2005; Müller and Padberg 2003). Simulation development has no major difference with developing any other software product and therefore developing techniques such as XP or TDD could be considered as tools which secure some level of accuracy in the implementation phase of simulation development.

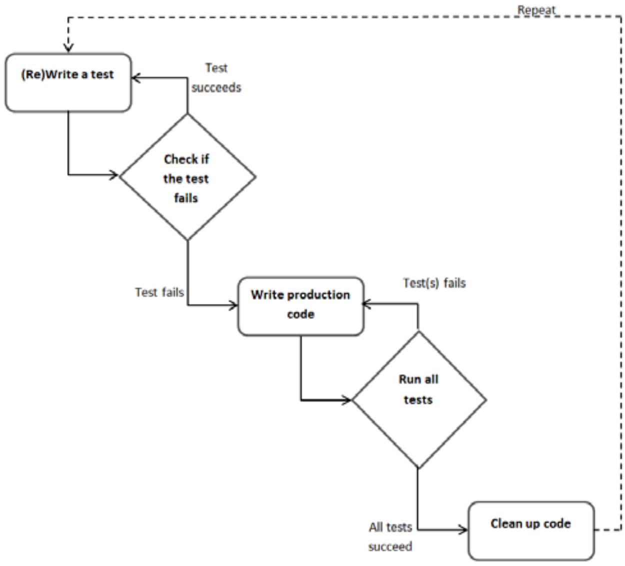


Figure 1 *Test-Driven Development Cycle*

In this study, we are particularly interested in the application of the TDD in Discrete Event Simulation (DES) on micro/macro levels. To this effect, we utilize a multi-paradigm visual interactive modelling and simulation (VIMS) environment, AnyLogic[1]. To the best of our knowledge this kind of approach has never been applied to developing Operations Research component based simulation models where researchers/practitioners often work with predefined components (design patterns) rather than coding methods.

We have conducted a feasibility study to investigate the opportunities of Test-Driven Simulation Development (for DES) within a VIMS environment. In our feasibility study, we simulate a surgical ward where the patients should spend no more than 30 minutes within the system. Our results show that it is possible to apply TDD to DES development in VIMS environment (combining coding the tests with the use of modelling elements in form of design patterns). Currently TDD deals with verification (comparing the conceptual model in form of requirements to the implemented model in terms of code) while the task description requires some form of validation (comparing the conceptual model with the real world). The rest of the paper is organized as the following. In Section 2 a brief survey of the researches conducted in this area is given followed by a description of our toolkit of choice (AnyLogic) in Section 3. Section 4, describes the case study (the NHS surgical ward problem) along with a detailed account of the application of TDD in the implementation phase of the simulation. Finally, conclusions are provided in Section 5.

2 RELATED WORK

This section provides the reader with a brief literature survey on the use of TDD concept in simulation. This research area is considerably new and the volume of work which can be referred to directly is scarce. One of the early studies on this topic was conducted in (Kleijnen 1995). In this study, the verification phase is divided into four steps: i) general good programming skills, ii) controlling intermediate simulation output by tracing, iii) comparing final output of the simulation with analytical results and iv) animation. The first step which was briefly described in (Kleijnen 1995) falls in the scope of our study. The author recommends a module-by-module testing of the simulation software development phase instead of *spaghetti* programming and point out that code analysts should *divide-and-conquer* the code verification process. Collier et al (2007), propose a TDD approach towards developing simulation. In this work, relying on the declarative structure of the simulation modelling toolkit that the authors have employed for their study, they show that a TDD approach is

[1] http://www.anylogic.com/

quite feasible. The toolkit in use is Repast[2] and it is argued that the reason why a TDD approach is possible is the fact that the toolkit maintains a strict separation between model and simulation concerns. That is, the coding can be performed isolated from the framework. An example of writing a code for the game Iterated Prisoner's Dilemma is given where a TDD approach has been taken to develop the model.

Zhang (2004), proposes the application of Test Driven Modelling to enhance the Model Driven Development process. The ideas from Test Driven Development are transformed and adapted for simulation modelling purposes. Our approach differs to the one proposed by Zhang (2004) due to the fact that instead of transferring the TDD ideas and adapt them for simulation modelling, we take a rather more direct approach by implementing the actual test units first instead of creating test models and replace the test units with them. Dietrich et al (2010), introduces a Model-Driven Simulation based on UML models. The unit testing method has been applied on UML simulation models, taking unit testing a level higher to the level of simulation models described in UML. The availability of automatic UML model transforming tools has reportedly enabled the authors to propose the idea. A new testing method is proposed in which UML is employed to specify model-level unit tests in order to validate simulation models defined with UML. The translation, execution and evaluation of the unit tests is described within the framework Syntony (Syntony is a standardized graphical modelling language, based on UML 2 which is proposed by Dietrich et al 2007). The proposed method allows the compilation and execution of the test code in combination with the simulation code. It has been shown that the proposed method provides a fully functional unit testing framework where the modeler relies on automated test case generation and execution.

3 THE ANYLOGIC TOOLKIT

AnyLogic is a recent and well known simulation toolkit. This toolkit consists of several libraries of objects which covers a wide spectrum of components necessary for designing different kinds of simulation paradigms (system dynamics, discrete event, and agent based). In AnyLogic, a model can be represented graphically while some Java code snippet can be added to implement the objects, component interactions and dynamics of the system. Though the implementation and modelling are not totally separate as it is the case with the Repast toolkit, the Anylogic toolkit is equipped with debugging and tracing utilities. An API library is also provided for users who wish to take a more advanced programming approach. Essentially, what the toolkit does is to assist the modeller to *write the minimum amount of Java code*. Based on these observations, and due to the fact that Anylogic follows an object oriented approach, one can easily conclude that TDD is feasible within the Anylogic framework. However, this claim is yet to be validated.

Since Anylogic is basically a graphical modelling environment, applying model-level unit tests as proposed by Dietrich et al 2010, seems to be well promising. Dietrich et al 2010 proposes to incrementally construct a simulation model by implementing model-level unit tests within the UML. However, it has been assumed that the underlying framework has the ability to automatically convert the UML model into executable code. But, utilization of this method, puts the development process into a new category, say, Model Driven Development (MDD). In this study, we apply TDD method for implementing a simulation model instead of deriving it. AnyLogic model is a tree of active objects encapsulating each other. The root of the tree is called the main object. The main object represents the highest abstraction level of the model. It embodies *Embedded Objects* (components), functions, variables and user defined objects. There are various libraries within the AnyLogic IDE which provide a wide range of components. We specially use the Enterprise Library as it contains the objects necessary to create simulation models. Each component comes with a set of events. For instance, events such as `onEnter`, `onExit` are triggered when an entity enters or leaves the component respectively. Each event has a field which gives the designer the opportunity to write a piece of code to determine the necessary action to be taken when the event occurs. This can be a single line of code (like increasing a counter), a function call or a class instantiation.

[2] http://repast.sourceforge.net/

Each component processes an entity. The user can define an entity class with various attributes and associate it to components. However, AnyLogic associates a default entity class (with no extra attributes) to each component. Indeed, AnyLogic provides various variables of a component with meaningful default values. This enables the users to use the components in a plug-and-play fashion without initializing them.

4 A CASE STUDY: NHS PROBLEM

4.1 Problem Description

The average waiting time in various wards within the National Health System (NHS) is nowadays a critical and much argued topic. In this problem domain, DES is a commonly used tool to help finding alternative solutions that (at least in principle) allow to maintain service standards while keeping costs low. For our case study we have chosen to build a model of a (fictitious) NHS surgical ward that could be used for improving the operation of the ward.

The Case: The NHS has recently opened a new walk-in centre. There have been several complaints about long waiting times for non-urgent patients in the surgical ward (which is part of the walk-in centre). In response to the complaints the surgical ward management has set itself the target to improve the service so that the overall time from arriving at the ward to the beginning of the treatment does not exceed 30 minutes for at least 95% of non-urgent patients. There are always at least 5 doctors present in the ward which has a capacity of 8 operating rooms. In order to achieve the targets regarding non-urgent patients, the surgical ward management is happy to employ additional doctors whenever necessary. Once patients arrive at the entrance of the surgical ward they have to register with the front desk. Then they can move to the waiting area. The surgical ward has to deal with non-urgent and urgent cases. If urgent cases arrive they will be allowed to jump the queue in the waiting area and will be treated as soon as a doctor is available (we assume that only one doctor is required for treating each patient). There is no problem with the availability of nurses to assist the doctors. In this study, we can safely assume that there are a sufficient number of nurses on the surgical ward to support any solution.

4.2 Simulation Implementation Using TDD

One of the objectives of DES is to keep track of the statistics of the system under simulation. That is, we are interested in tracking the changes of the system status, triggered by occurrence of an event, right from the moment when an entity enters the system, until the moment it leaves the system (Nance 1993; Shannon 1998). Based on this definition and rephrasing the NHS ward example accordingly, an event is triggering the arrival of a new patient and the statistic that we are interested in collecting the average waiting time of many such patients (events). Thus, this definition suggests that the two major components we should consider as the starting point in system modelling are events, entities and systems entry and exit. We start with the entity, without which having a system is pointless.

It is obvious that an entity (a patient in our case study) needs to be represented as an object (a class). Based on TDD approach, the first thing that needs to be done is to design a test. This test can be in form of a test class or a test function. Judging by the simplicity of our case study, a test function is chosen to be written instead of a test class. The test function `testPatientClass` (located in the main class) can be seen in Program 1. It tests the `patientClass` (which currently does not exist because it is not written yet) by checking the registration and urgency status of the patient. The type of the variable `patient` in Program 1 is `patientClass`. Please note that, this is our strategy throughout the paper that prior to implementing a class a test function or a test class is written first. As a standard, the name of this test function/class is the same name for the intended class with an additional *test* prefix.

Running the simulator produces an error due to the fact that first, the class `patientClass` does not exist and boolean parameters `isUrgent` and `isRegistered` are not available. In order to resolve the issue the class is designed with the two boolean variables mentioned above. This time simulator is run without errors. The `patientClass` in its current form can be seen in Program 2. From this point onward, the input/output of every component which is added to the model is set to the instance of the `patientClass` instead of the default value which is `Entity`. In AnyLogic, the input/output of a component determines the type of the entity which can be a user defined class or the default type `Entity`. Of course, in case this is forgotten, running the simulation will result in run-time errors.

As discussed earlier, based on the DES definition, we need to consider the second component. Entrance and exit points of the system are marking points in which the system status changes. That is why, in the next step, we realize that the system requires a source (system entry) and a sink (system exit) where the number of incoming customers has to be equal to the number of the customers leaving the system. This constraint is again inferred from the DES definition where it highlights the fact that an entity which enters the system is bound to leave it at some point. On the other hand TDD spirit suggests that given a constraint (or any other design component for that matter) a test has to be written first. Thus, a test function located in the main class is written (Program 3), which examines this constraint. The test, namely `test_inOutMatch` calls the function `inOutMatch` which checks for a match between the number of customers in and out the system. Calling the test results in error, obviously because the function `inOutMatch` does not exist. This function is then written as in Program 4. Again, running the simulation results in an error since there is no source and sink components in the system. Consequently, a source and a sink are added from the Enterprise library and connected to each other. This time the simulator runs smoothly.

Program 1 testPatientClass

```
boolean ir = patient.isRegistered;
boolean iu = patient.isUrgent;
return ir&&ru;
```

Program 3 test_inOutMatch

```
if (inOutMatch() == false)
        error("in-out count does not match");
```

Program 4 inOutMatch

```
if (source.count() == sink.count())
        return true;
else
        return false;
```

Program 2 patientClass

```
public class patientClass
{
        public boolean isUrgent;
        public boolean isRegistered;

    public patientClass()
    {
        isUrgent = false;
        isRegistered = false;
    }
    @Override
    public String toString()
    {
        return super.toString();
    }
}
```

Now that the simulation model is initialized through the addition of a source and sink, we can move on to the next step, that is, measuring the *time in the system* of all the patients. This is a hard constraint and we always need to check it to see if it exceeds the 30 minutes (0.5 hour) threshold for 0.95% of the patients in the system. This is why the test function `testOverwaiting` (Program 5) is written. and called in the *on enter* event of the sink component. `testOverwaiting` is the acceptance test. The validity of the entire system depends on passing this test. Running the simulation produce an error since there is no time measurement component called `tmEnd` in the system. Running the simulator produces yet another error. This time the reason is that there is no component which indicates the start of the time measurement. Such a component (`tmStart`) is added subsequently. The model in its current form is shown in Figure 4.

The very existence of a registration desk suggests that the subsequent departments/units within the surgical ward require some initial information from a patient which should be acquired during the registration process. That information includes specifics, such as, whether a patient is urgent or not. Thus, in this step we add the registration desk to our model and write some tests to check the registration of a patient as well as his/her urgency. The registration desk consists of a service component connected to time tracking components from both ends. A third connection is also provided by which the registration desk is connected to a resource unit, say, the registrar. What we need to make sure is to assert that every patient is unregistered prior to entering the registration desk and otherwise after leaving the desk. Thus two test functions are written in the main body of the simulation project to check such conditions (Program 6 and Program 7). Please note that the first test (`testDeskEnteringPatient`) is checked in the *on enter* event of the registration desk service component while the second test (`testDeskLeavingPatient`) is checked in the *on exit* event.

Program 5 testOverwaiting

```
int overwaitingCounter = 0;
int totalCounter = 0;
double sum = 0;
for(int i=0;i<tmEnd.dataset.size();i++)
{
        if(tmEnd.dataset.getX(i) > 0.5)
                overwaitingCounter ++;
        totalCounter ++;
}

if ((double)overwaitingCounter/
    (double)totalCounter > 0.95)
{
        error("over waiting !!!");
        return false;
}
return true;
```

Program 8 inOutMatch

```
int s=0;
s+=registrationDesk.queueSize();
s+=registrationDesk.delaySize();
s+=sink.count();
if (source.count() == s)
        return true;
else
        return false;
```

Program 6 testDeskEnteringPatient

```
if (patient.isRegistered() == true)
{
        error("A patient is registered
               without attending the
               registration desk!");
        return false;
}
return true;
```

Program 7 testDeskLeavingPatient

```
if (patient.isRegistered() == false)
{
        error("A patient is leaving
               the registration desk
               unregistered!");
        return false;
}
return true;
```

Running the simulation results in an error which is triggered in the second test (Program 7). The reason is that while serving the patients at the registration desk, their registration status is not changed. Therefore the necessary code is written to pass the test. That is, the code line `patient.registered=true;` is added to the *On enter delay* event of the `registrationDesk`. Running the simulation once more triggers another error in the function `test_inOutMatch`. The reason being that, the service component introduces delay to the system by putting some entities in its internal queue. Hence the number of entities in the sink is always less or equal to that of the source. At this stage we need to amend the `inOutMatch` to pass the `test_inOutMatch` (Program 8). After this extension to our test function, the simulation runs without producing any errors (Figure 5).

In the next step, we need to check if the patients are split according to the urgency of their cases. Whether a patient is in need of urgent attention or not is decided upon by the registration desk. While urgencies are dealt with without a delay, non-urgent patients need to wait in the ward. A queue presents this waiting area for non-urgent patients. This is why a queue should be added to the model. However, prior to adding the queue, a selector (`urgencySelector`) is needed to split the patients into urgent and non-urgent categories. At this point, based on our TDD approach, we implement a test in which proper checks are made to verify the existence of the components as well as their functionality. The first test, `testSelector`, would be regarding the first component we intend to add, that is the `urgencySelector`. It simply checks whether the selector distributes the entities according to the stated requirements (Program 9). Running the simulator triggers an error (via `testSelector`) since we have not added the selector component. Thus, in order to pass the test we add this component (`urgencySelector`) and set its probability to 0.05. The `true` port of the selector corresponds to urgent cases whereas the `false` port corresponds to non-urgent ones. Both ports are connected to the `tmEnd` component. The probability 0.05 addresses the probability of entities which are assigned to the `true` port (urgent patients). Running the simulator at this point does not generate any errors.

The next entity is the waiting queue (`waitingArea`) for which we need to write a test. However, this time, instead of writing a new test, we decide to extend an existing test function `test_inOutMatch`. This test function relies on the function `inOutMatch` which is now modified as in Program 10 and includes the number of entities in the queue and the delay service of the `waitingArea`. Running the simulator produces an expected error, notifying that the component `waitingArea` does not exist. This component is added between the `false` port of the `urgencySelector` and the `tmEnd` component. Figure 6 shows the latest modifications on the model.

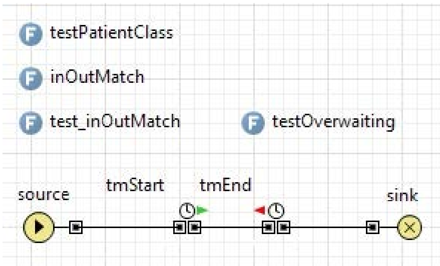


Figure 4 *Addition of time tracking components*

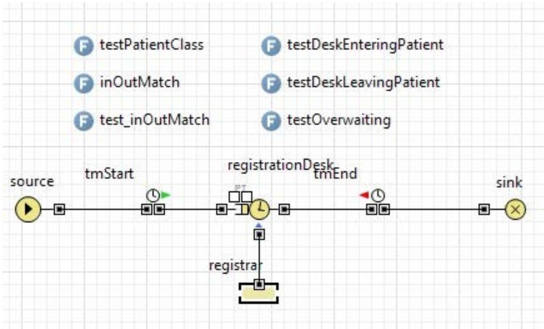


Figure 5 *The registration desk*

In our next step, we need to proceed with the implementation by adding a resource component to the system which represents the available doctors by which both urgent and non-urgent patients are served. However, the urgent patients always have higher priority than the non-urgent patients. That is why a secondary queue which is a priority queue (`priorityQueue`) should be added to the model. To implement a test for this new component, once again the `inOutMatch` function is modified to test the functionality of the new component by adding the number of entities in the `priorityQueue` to the overall number of entities within the system. Running the simulator produces an error indicating the absence of the `priorityQueue` component which is added subsequently and placed before the `tmEnd` component. The priority of each entity is set to be the value of the Boolean variable `isUrgent`. Subsequently, and according to the problem description where we initially have 5 doctors and 8 rooms, the service component (`ward`) should be added to the model. Similar to the process of adding the `waitingArea` and `priorityQueue` components, we extend the `inOutMatch` function to test the new component (Program 11). After running the simulation and receiving the error, notifying the absence of the component we add the `ward` component between `priorityQueue` and `tmEnd` (Figure 7).

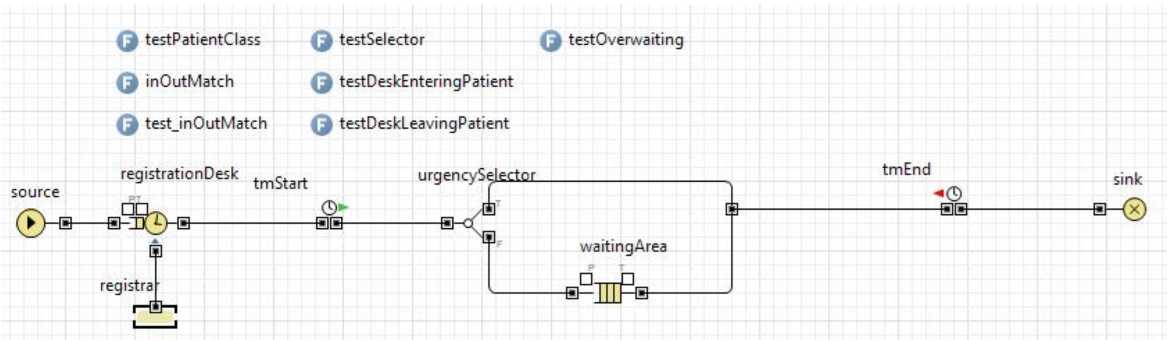


Figure 6 *Adding the urgency selector and the waiting area to the model*

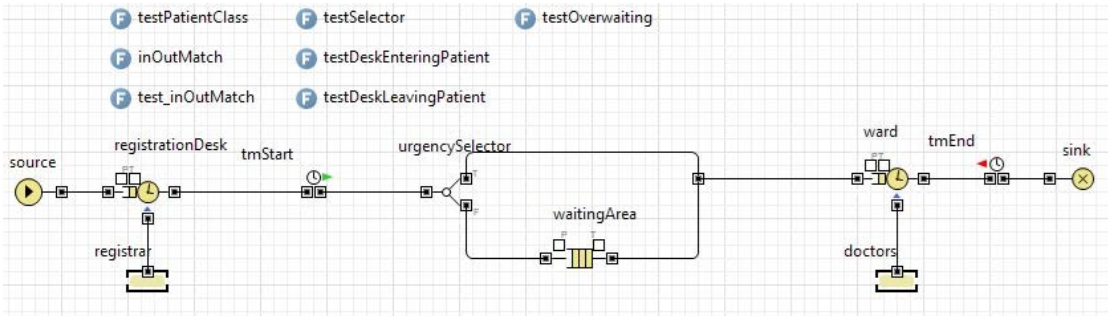


Figure 7 *The tmEnd component is in a wrong place causing testOverwaiting test function to fail.*

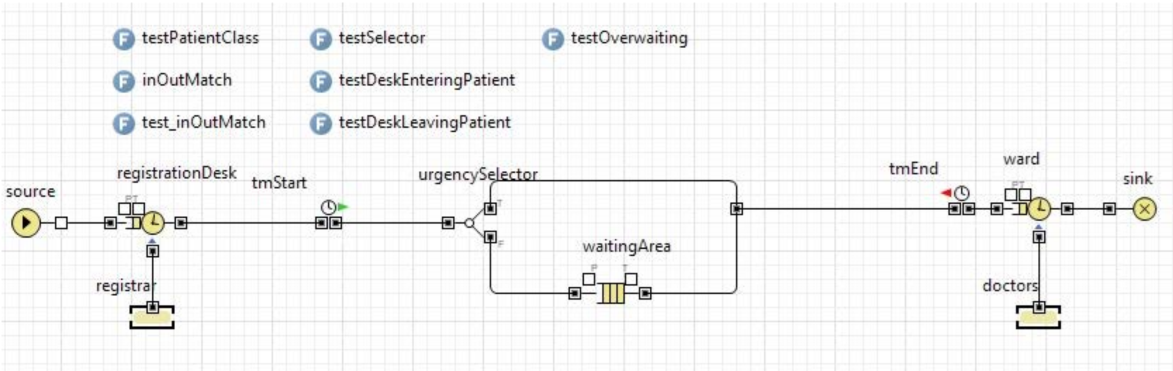


Figure 8 *The final system*

Running the simulation produces a very interesting error. The error is generated by the test function `testOverwaiting`. The reason is that the time measurement components are placed in the wrong place where they also cover the time spent by the patient while he/she is being attended. That duration certainly should not be included in the waiting time. Hence, the `tmEnd` component which measures the time a patient leaves the waiting area is shifted leftwards and placed before the `ward` service component. Running the simulation after these modifications does not produce any error any more. Figure 8 shows the final model. Although our case study in this paper consists of a small DES project, one can easily note that the process is simple and rather repetitive. Once a set of *standard* unit tests are established for each component of a DES system, one can automatically run the *standard* unit tests prior to adding a component to the model. For instance, it is clear that `test_inOutMatch` is necessary for a pair of source and sink components regardless of the conceptual model.

Program 10 inOutMatch

```
int s=0;
s+=registrationDesk.queueSize();
s+=registrationDesk.delaySize();
s+=waitingArea.size();
s+=sink.count();
if (source.count() == s)
        return true;
else
        return false;
```

Program 9 testSelector

```
p=urgencySelector.probability(entity);
if (p != 0.05)
        error("wrong probability!");
```

Program 11 inOutMatch

```
int s=0;
s+=registrationDesk.queueSize();
s+=registrationDesk.delaySize();
s+=waitingArea.size();
s+=priorityQueue.size();
s+=ward.queueSize();
s+=ward.delaySize();
s+=sink.count();
if (source.count() == s)
        return true;
else
        return false;
```

5 DISCUSSIONS AND CONCLUSION

In this paper, we have demonstrated that a TDD approach can be used to develop and implement a DES model. We have described the development process step by step using a case study. Our approach can be summarized in the following guideline:

1. Pick a requirement, i.e., a story
2. Write a test case that will verify a small part of the story and assign a fail verdict to it
3. Write the code that implement particular part of the story to pass the test
4. Execute all tests
5. Rework on the code, and test the code until all tests pass
6. Repeat step 2 to step 5 until the story is fully implemented

There is some evidence that while developing a DES in a VIMS environment using a TDD approach, one can extract test patterns specific to the components that the project requires. Each DES component (such as sink, source, queue etc.) can be associated to a group of unit tests that should be called prior to adding the component to the project. While we do not necessarily assume an identical identity for programmers and modellers, the test patterns will help both groups to develop DES models faster and in a more reliable fashion. Furthermore, such test patterns help in employing the TDD approach in VIMS environments such as Simul8 and Witness where much less coding effort is foreseen.

We have employed the TDD approach at two levels, namely unit testing and acceptance testing. We observed that applying TDD at unit testing level in relation to DES is straight forward, but not for acceptance testing. Considering the case study we have used, while the final model (see Figure 8) runs smoothly without producing any immediate error, the test function `testOverwaiting` may still generate an error during a longer run. This is due to the fundamental difference between developing a simulation model and an ordinary software model. In the latter case, the objective is to develop a product which is supposed to function perfect. However, in case of developing simulation models, the objective is to design a product where imperfections of the system under study are outlined.

Moreover, prior to constructing the model, no attempts were made to acquire information regarding the probability distribution of the arrival of the patients, the attendance duration and etc. This does not mean that no such distribution was used in the model constructed above. Such distributions exist and AnyLogic's default distributions (along with their default parameters) were used without any change. The reason for such an approach stems from our test first method. In TDD, minimal coding is essential to the success of the approach. Making an analogy, we decided to extend that idea for the Test Driven Simulation by employing minimal data/knowledge. The purpose is to check if the available test can help to expose the missing data/knowledge of the system. As it happens, it does indeed. It is interesting to point out that this test was actually the second test written for the system and when it was written the only components of the system were the source and the sink. It is also interesting to emphasize the fact that when one tries to resolve the issue, model modification and model parameters (instead of coding) is required to make the test pass. In other words, instead of writing minimal code to make the test pass, providing an additional, though minimal, knowledge about the system is essential to make the test pass. Evidently, it is possible to take our approach to a higher level of abstraction where the unit tests contribute to the specification of model parameters. If the simulation model is given sufficient time, it forces the designer to introduce some additional parameters (minimal knowledge to make the test pass) with which the simulation passes the test.

As a result, we have presented a method to apply the TDD philosophy to simulation development. Whether or not the discussions above hold in general is a part of what we intend to investigate further as a future work.

REFERENCES

Beck K, Beedle M, Van Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, Highsmith J, Hunt A., Jeffries R, Kern J, Marick B, Martin R C, Mellor S, Schwaber K, Sutherland J and Thomas D (2001). Manifesto for Agile Software Development.

Beck K (2002). Test Driven Development: By Example. *Addison-Wesley Longman Publishing Co., Inc.*, Boston, MA, USA. 2002.

Bhat T and Nagappan N (2006). Evaluating the efficacy of test-driven development: industrial case studies. *In Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering (ISESE '06)*. ACM, New York, NY, USA, pp.356-363.

Borshchev A, Simulation Modeling with AnyLogic: Agent Based, Discrete Event and System Dynamics Methods.

Collier N T, Howe T R, and North M J (2007). Test-Driven Simulation Development using Repast Simphony, *Proceedings of the North American Association for Computational Social and Organizational Science (NAACSOS) 2007 Annual Conference*, Emory University, Atlanta, GA USA.

Dietrich I, Dressler F, Schmitt V and German R (2007). SYNTONY: network protocol simulation based on standard-conform UML 2 models. *Paper presented at the meeting of the VALUETOOLS*.

Dietrich I, Dressler F, Dulz W, and German R. (2010). Validating UML simulation models with model-level unit tests. *In Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools '10)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, , Article 66 , 9 pages.

Erdogmus H, Morisio M and Torchiano M (2005). On the effectiveness of the test-first approach to programming, *IEEE Transactions on Software Engineering*, vol.31, no.3, pp.226-237.

Gelperin D and Hetzel W (1987). Software Quality Engineering, *Proceedings of Fourth International Conference on Software Testing*, Washington D.C.

Kleijnen, Jack P C (1995). Verification and validation of simulation models, *European Journal of Operational Research*, Elsevier, vol. 82(1), pp.145-162.

Kleijnen Jack P C, Bettonvil B, and Van Groenendaal W (1998). Validation of Trace-Driven Simulation Models: a Novel Regression Test. *Manage. Sci*. 44, 6 (June 1998), pp.812-819.

Larman C and Basili V R (2003). Iterative and incremental developments. a brief history, *Computer* , vol.36, no.6, pp.47-56.

Müller M and Padberg F (2003), About the Return on Investment of Test-Driven Development, *In International Workshop on Economics-Driven Software Engineering Research EDSER-5*.

Nance R E (1993). A history of discrete event simulation programming languages. *In The second ACM SIGPLAN conference on History of programming languages (HOPL-II)*. ACM, New York, NY, USA, pp.149-175.

Newkirk J W and Vorontsov A A (2004). Test-Driven Development in Microsoft .Net. *Microsoft Press*, Redmond, WA, USA.

Shannon R E (1998). Introduction to the art and science of simulation, *Simulation Conference Proceedings*, 1998. Winter , vol.1, pp.7-14.

Zhang Y (2004). Test-driven modeling for model-driven development, *Software, IEEE* , vol.21, no.5, pp.80-86.

AUTHOR BIOGRAPHIES

SHAHRIAR ASTA received a BSc Computer Engineering from the University of Isfahan in 2003. In 2012 he received his MSc Computer Engineering from Istanbul Technical University. He is currently a PhD student in the Automated Scheduling, Optimization and Planning (ASAP) group at the University of Nottingham (http://www.cs.nott.ac.uk/~sba).

ENDER ÖZCAN is an Operational Research and Computer Science lecturer with the ASAP research group at the University of Nottingham (http://www.cs.nott.ac.uk/~exo). He was appointed to the EPSRC funded LANCS initiative and has been leading the Systems to Build Systems research cluster. He has been awarded grants as PI and co-I from a variety of funding sources. He has co-organised many workshops streams on hyper-heuristics/metaheuristics and has many publications in the area. Dr Ozcan is an associate editor of the International Journal of Applied Metaheuristic Computing and Journal of Scheduling.

PEER-OLAF SIEBERS is a lecturer in the School of Computer Science at the University of Nottingham in the United Kingdom (http://www.cs.nott.ac.uk/~pos/). His main research interest is the application of data driven computer simulation to study human-centric complex adaptive systems. This is a highly interdisciplinary research field, involving disciplines like social science, psychology, management science, operations research, economics and engineering. Other areas of interest include Risk Analysis and Systems Biology.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

A DISCRETE EVENT SIMULATION FOR THE ANALYSIS OF THE HARVESTING, TRANSPORTATION AND PROCESSING SYSTEMS OF A SEASONAL VEGETABLE PRODUCTION OPERATION

Dr Nicky Yates

Cranfield School of Management, Cranfield University,
Cranfield, Bedfordshire
nicky.yates@cranfield.ac.uk

ABSTRACT

This paper describes the development of a discrete event simulation model of a vegetable production operation from harvesting to bulk freezing of the product. The operation is complex and time pressured, the harvesting season is only a matter of weeks and the time from harvesting to freezing is measured in hours. Ensuring a smooth flow of product to the processing lines and maximising availability of the lines is therefore vital. The model aimed to provide a low risk environment for improving understanding of the current process and testing potential improvements. The results led to significant improvements in the understanding of the impact of field moves, breakdowns and processing different varieties on the operation and led to a number of recommendations which have increased the efficiency and reliability of the operation. In addition a 3 day model was built to facilitate decision support during the season.

Keywords: Harvesting, Processing, Logistics, Agriculture, Vegetable

1. INTRODUCTION

The harvesting and processing operations of agricultural perishable food products such as fruits and vegetables are often complex and time pressured (both in terms of processing time and time when the crop is available). The situation discussed in this paper is just such a process, involving the production of over 1000 tonnes of a frozen vegetable product every day using five geographically dispersed harvesting groups and five processing lines on two sites. The season of harvest of the vegetable is short at only eight to ten weeks per year and the sales of the product produced constitute a significant proportion of the bottom line of the case company. This means that the success of the harvesting operation is vital and the constituent processes are subject to detailed and carefully managed planning. The operation is already well run with a large number of sophisticated planning processes in place to ensure that the vegetable product is harvested to facilitate a smooth flow of product into the factory. This maximises the productivity of the processing lines which are maintained to minimise downtime. However, the time constrained nature of the operation means that the opportunity to experiment and try new things to improve the operation further is limited, as anything which could jeopardise the success of the operation is treated with justified caution. Therefore, simulation was proposed as a low risk way of increasing the understanding of the interactions which occur in this complex process and to test a number of possible improvements prior to attempting them in reality.

The crux of the problem is to ensure that a uniform feed of vegetable is supplied to the process lines. This is not the first paper to address such an issue, the uniform feeding of sugarcane, orange and wood production operations has been addressed previously by work such as Iannoni and Morabito (2006) , Neves *et al.* (1998) Martin *et al.* (2002) Higgins and Muchow (2003). In this operation a critical constraint is that of time, Semenzato (1995), Arjona *et al.* (2001) and Neves *et al.* (1998) discuss that once harvested sugarcane and oranges need to be processed within a limited period of time to preserve their quality, in these cases this time is measured in days. For the case considered here the time is measured in a small number of hours, thus further complicating the situation. The short time period means that practically no buffer exists within the system, if product is not fed smoothly to the line then either queues will build up at the factory and vegetable will not be processed within the specified time and will be downgraded or the line will not be fully utilised reducing operating capacity for a given day. In addition if breakdowns occur and the lines are not available vegetable which is harvested will not be processed in the required time and will also be downgraded. A number of previous studies have successfully used discrete event simulation to analyse logistics systems in the agro industries these include Mathew and Rajendran (1993), Semenzato (1995), Bradley and Winsauer (2004) and Iannoni and Morabito (2006), Andrieu *et al.* (2007). The majority of these studies address the issue of supply to the factory based on models of harvesting and transportation operations only, for cases where time constraints are significantly less onerous than in the present case. This work extends previous work by also including the processing operation in the model. This allows the impact of breakdowns and line assignments for processing of multiple products to be taken into account, both of these can have a significant impact on the current case particularly due time sensitivity of the product.

1.1. Research Questions

The work set out to answer the following research questions:
RQ1: What are the factors which have the most impact on the ability of the vegetable operation to run efficiently?
RQ2: How can the way the vegetable production operation is managed, be improved to make it more efficient and thus reduce costs?

Simulation was adopted as it provided a low risk method for increasing understanding of this complex process. The major advantage of simulation here is the ability to carry out a systematic set of experiments changing a number of factors known to be important to the running of the operation in a known manner and hence quantify their impact. This would be practically impossible to do in practise. The factors to be assessed were identified as part of the extensive data collection study prior to building the model. A series of interviews with staff working in different parts of the operation were carried out. As part of these interviews participants were asked what factors they believed to be important to the efficiency of the operation. The most common responses were the impact of breakdowns (both production lines and harvesters), the impact of the second variety on the operation and the impact of moving the harvesters around the regions (field moves). A set of experimental runs for each of these factors was designed to assess their impact in a systematic manner. Due to lack of time although validation runs were repeated to determine impact of variability of the input parameters on the output of the model, this was not possible for the experimental runs.

2. THE MODEL

The aim of the model was to minimise the total supply chain cost of producing the required tonnage of frozen vegetable with the right quality. The vegetable harvesting operation studied planned to harvest and process 40,000 tonnes of vegetables in 40 days. Harvesting is achieved using 5 geographically dispersed harvesting regions, each with 3 harvesters. The vegetable is then transported to the factory using a fleet of up to 30 wagons, where it is processed using 5 processing lines. The vegetable produced is made up of two grades which must be processed separately and all vegetable must be processed within 2.5hrs. Figure one shows a simplified version of the model for one region and one processing line for illustration.

The model was developed using WITNESS discrete event simulation software supplied by the Lanner Group. The data on which the model is based was collected during two harvesting seasons, cleaned and analysed before incorporation in the model. The following section describes the constituent elements from which the model is composed.

2.1. Harvesting

The harvesting operation models the harvesting of the vegetable including moving harvesters between fields. It is composed of the following elements (the part is *Fields*):

Field Buffer - Holds all the regions fields. Each field is assigned attributes e.g. hectares, yield, journey time based on historical data.
Harvest Queue – A conveyor which holds the next 4 fields to be harvested, *Fields* are pulled out of *Field Buffer* in harvest order.
Field Splitter – A machine which simulates the movement of harvesters between fields. In reality the fields to be harvested lie over a large geographical area, the main constraint being that they must be within an 80 minute drive of the factory. It would be possible using Witness, to model the complex network of fields and roads which serve the area, however, this is unnecessary. The fields to be harvested in any day are considered as a pool and the *Field Splitter* machine with variable cycle time is used to simulate the time taken to move between fields. In addition the machine splits the field tonnage (yield * hectares) into loads for harvesting.
Harvest Buffer – A buffer which holds all the harvester loads for the current field.
Harvester Buffer Feed - Machines which contain the logic for feeding the *Harvesters*. Essentially each machine pulls a harvester load out of *Harvest Buffer* at the time dictated by the harvest programme as long as the *Harvester* is not broken down and there are sufficient loads in the *Harvest Buffer*.
Harvesters - 3 machines each with a 15 min cycle. *Life* (an attribute which is used to ensure that all the vegetable is processed within the specified time) is assigned to an attribute on entry. The *Harvesters* breakdown – time between breakdowns is modelled using a negative exponential distribution with a mean of 4.5 days, the breakdown time is modelled using a lognormal distribution with mean of 60 minutes. Times are based on historical data.
High Lift – Machine which takes 3 *Harvester* loads and converts them to a vegetable unit for transport to factory. Cycle takes 3 minutes. An additional buffer acts as a 2nd High lift if there is no lorry on the *Field Load* track.

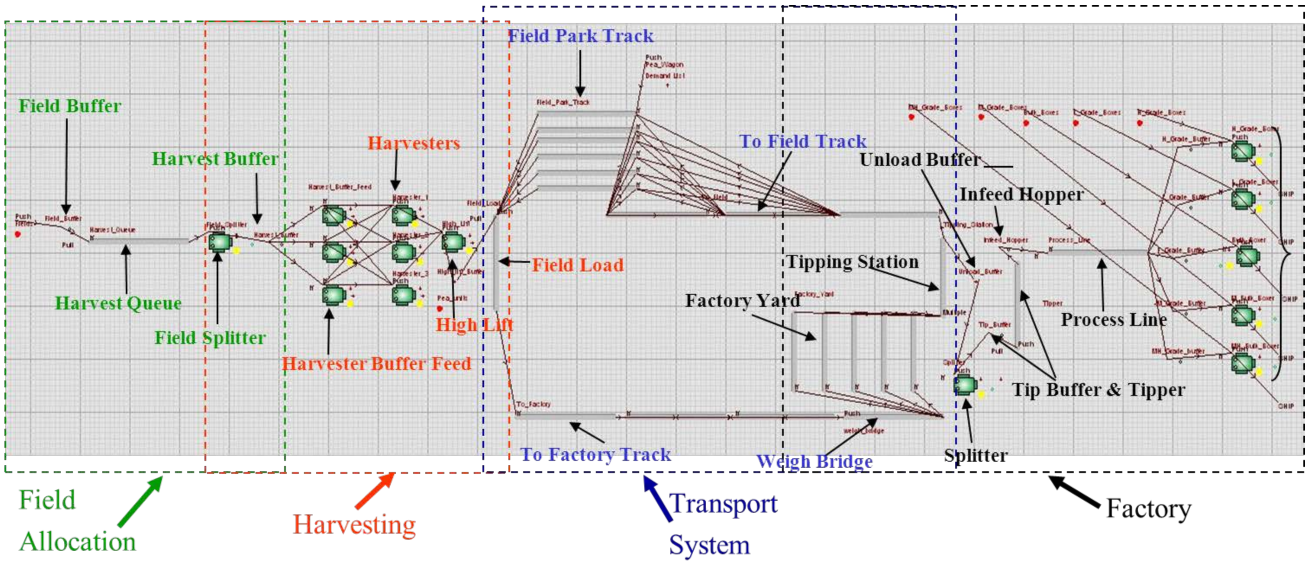


Figure 1 *Simplified Witness Model for one harvesting region and one processing line*

2.2. Transport

This models the transport operation from the field to the factory and is composed of the following elements:
Field Load - Wagon loading track. Wagons wait on the track until 2 loads are present unless the model dictates otherwise. Loading takes 2 minutes.
Field Park Track - Essentially models wagons parked in the field. Wagons exit to *Field Load* when harvested vegetable enters the High Lift. Exit rule is dictated by the harvest programme.
To Factory Track - Takes wagons from field to factory. Transit time dictated by journey time attribute of the field.
Weigh Bridge- Models the factory weighbridge operation. Residence time 10 minutes

Factory Yard - Allows 5 wagons to queue in the factory. Includes coding to ensure wagons enter the factory in the order which maximises good vegetable.
To Field Track - Takes wagons from the factory to the field. Transit time is dictated by the journey time attribute of the field that the wagon is returning to.

2.3. Processing

This models the processing operation in the factory and is composed of the following elements:
Splitter - Splits the vegetable unit from the wagon into 50kg units (+ 1 smaller unit) for processing.
Tip Buffer & Tipper - Models the continuous tipping of each load, total time 3 minutes.
Infeed Hopper - Infeed hopper for process line. When empty a check is performed to determine if a changeover time is required for changing the lines between the vegetable types.
Unload Buffer - Holds unloaded vegetable units from the wagon. The 2nd unit is pushed to the tipper when the *Infeed Hopper* contains less than 3 units. On exit proportions of rough and damaged (waste) are calculated and the weight attribute adjusted to paid weight.
Tipping Station - Unload track for wagons. Vegetable unloads into the *Unload Buffer* but the Wagon cannot exit until the 2nd load is tipped into an infeed hopper.
Process Line - Residence time 20 minutes capacity 14 tonnes/hr. On input the weight attribute is adjusted for factory yield. On output the variety and the "life" of the vegetable is determined and from this the grade established. *Process Line* has two types of breakdown: (i) Sanitations - planned stops accounted for in the harvest programme, these take place every day for an average of 2hrs (based on distribution) (ii) Breakdowns – not accounted for, based on distributions derived from historical factory data, harvesting is stopped on breakdown and restarted on repair.
Boxing System - On exit from *Process Line* vegetable units are sent to the appropriate buffer for their grade. The weight of vegetable in each buffer is monitored, when it reaches 1 tonne the vegetables are sent to the boxing machine. The total mass of each grade and the total rough, damage and yield losses are recorded.

2.4. Modelling Challenges

Taking the model from a one region one processing line prototype model to the full multi region multi process line model was a complicated and lengthy process involving a large number of challenges. An example of this is modelling the targeting operation which determines the load sizes to be harvested in each region. Initially these values were fixed, however this does not reflect what happens in reality where the load size is calculated dynamically for each harvesting period based on: the yield of the field, the requirements of the factory and the status of other regions. Using fixed periods would lead to oversimplification in the model and the model not reflecting many of the challenges faced by the real operation, the targetter in the model was therefore updated to reflect this, a simple task for a single region, but extremely challenging for the full multi-region model due to the complex interactions which exist.

2.5. Validation

Model validation was an extensive and time consuming exercise. All collected data was analysed and then sent back to the data originator for data validation. A prototype model of a single region and single processing line was built to test the concept. This was validated using face validity by allowing the agricultural manager (key contact with the most knowledge of the operation) to observe the model running and examine model outputs. The model outputs were also validated against historical data and the agreement was found to be good. This allowed the project to move to the next phase and the full model was built. During this process regular meetings were held with the operational management team to update on progress and also to discuss the appropriateness of assumptions and changes to the model and the modelling approach. When complete the model was run for a whole season using historical data for the fields harvested on each day, as input data allowing a direct comparison of the output with historical harvest data. This output was first used to identify days when the model behaved very differently to reality. The model was then observed allowing identification and correction of a large number of errors in the model, both in terms of errors in the model code itself and inconsistencies in the assumptions and model definitions. The results of the final validation and the model itself were shared with operational management team who agreed the model provided a realistic representation of the system and was therefore shown to have face validity.

Final validation results compared to historical data from the operation (telex) are shown in figure 2. Good agreement between model and reality was found particularly during steady operation. On a cumulative basis the model produced output 6% greater than the operation in reality. Once the model was validated the variation within the model itself was determined by running the model 5 times using different random number streams. On a cumulative basis the total tonnes varied by ±0.02% on average over a full run and the good vegetable by ±0.3%, variation within the non-good vegetable was greater at ±2%.

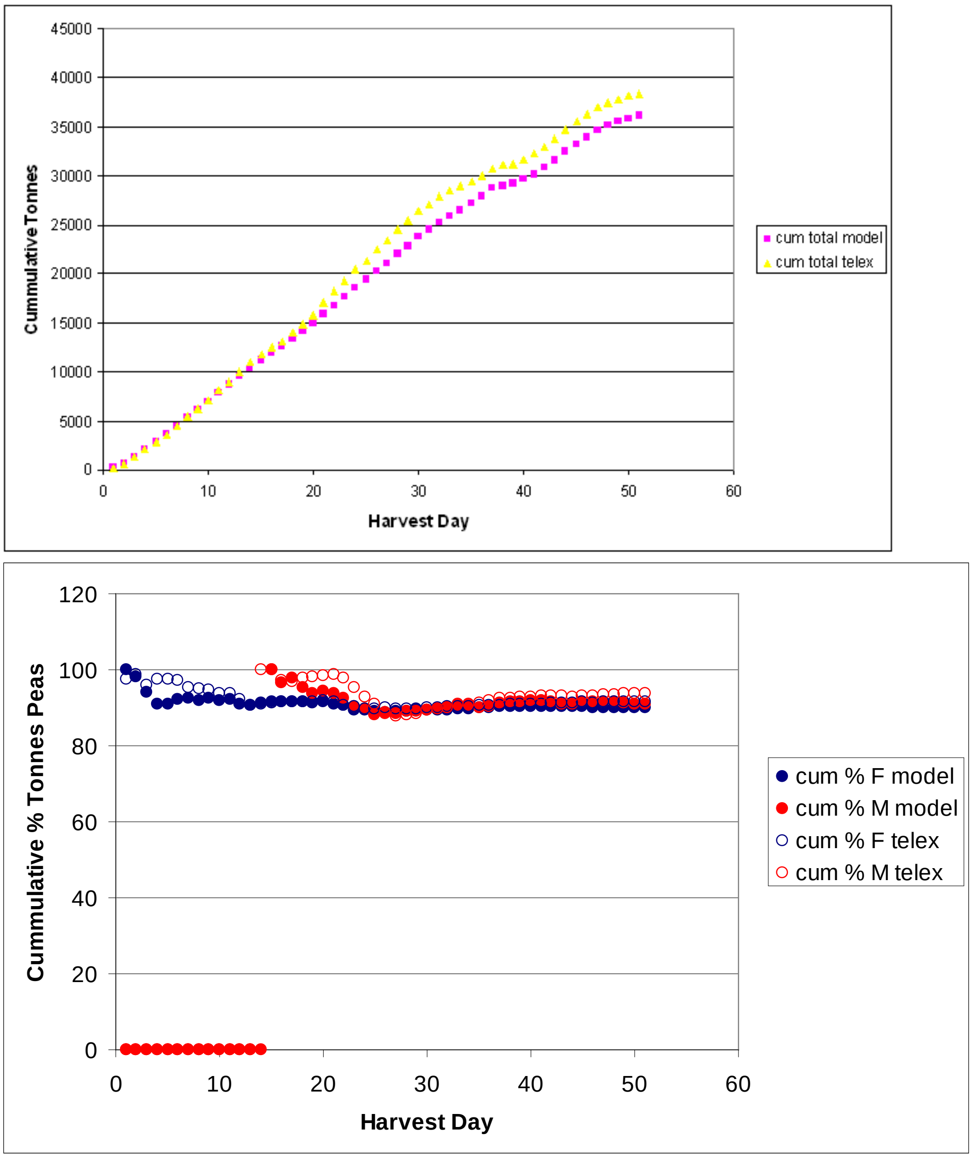


Figure 2 *comparison of simulation output with historical data*

3. MODEL RESULTS

Following successful completion and validation of the model extensive experimentation was carried out. Two key performance parameters were used for comparing model results. *Total "F" Grade Tonnes in 48 hr.* This is total tonnes processed by the model in 48hrs multiplied by the cumulative %

of good vegetable. *% No Vegetable Time after 48 hrs.* This is the % of the time that factory lines have no vegetable to process, corrected for the entry effect at the start of the run.

3.1 The impact of breakdowns

A series of model runs were carried out to determine the impact of processing and harvester breakdowns on the operation, the results are shown in figure 3 (closed symbols). In 48 hrs. The standard model produced 1973 tonnes of frozen F grade vegetable, this is 79% of the theoretical output. The no vegetable time for this period is however only 5.5%. The impact of removing harvester breakdowns is relatively small, in fact the number of tonnes of good vegetable produced are reduced by 2%. This is probably due to the fact that with no harvester breakdowns there is more vegetable on the road when a process breakdown occurs and therefore more slightly delayed vegetable is produced. No vegetable time is reduced by 0.8% due to a more consistent flow of vegetables into the factory. Removing process breakdowns has a much more significant impact on the output of the model. The tonnes of F grade vegetable increases to 2344 tonnes, 94% of the theoretical output and the no vegetable time is reduced by 3.2% to 2.2%. No Harvester or Process line breakdowns maximises the good vegetable produced and minimises the no vegetable time as would be expected. The tonnes of good vegetables produced however only increases slightly (~ 0.2%) confirming both that process breakdowns have the most significant impact on the operation and that the harvester breakdowns actually introduce some flexibility into the model. The no vegetables time reduces by a further 0.7 % to 1.5% of the available time, again illustrating that a more consistent flow of vegetables into the factory is achieved.

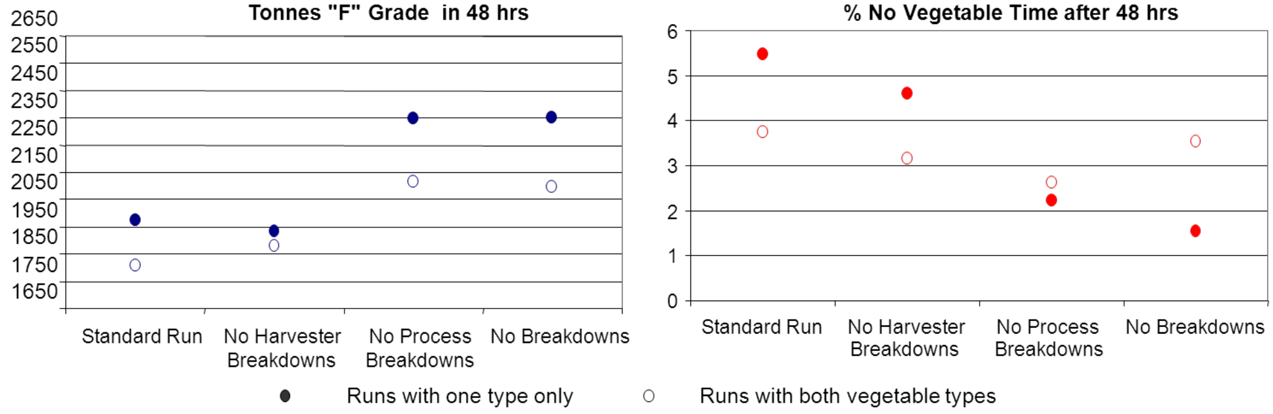


Figure 3 *The impact of breakdowns and processing both varieties of vegetable on the operation*

3.2 Introducing the second vegetable variety

It is known that when both varieties of vegetable are harvested the situation becomes much more complicated. The no breakdown runs were repeated when both types were processed; the results are shown in figure 3 (open symbols). Introducing the second vegetable type decreases the output of the model in all cases (on average by ~ 8%). The output of the model processing both types of vegetable increases slightly when harvester breakdowns are removed, but process breakdowns again have a more significant effect on output. The impact of removing breakdowns on the no vegetables time is limited reflecting the increased complexity in the operation when both types of vegetable are being processed.

3.3 Increasing the number of fields

Field moves are known to have an impact on the operation, as harvesting cannot occur when the harvesters are moving between fields. The impact of this lost time was investigated by increasing the number of fields harvested within the time period. The results are shown in figure 4 for the standard run. For one type of vegetable the output is maximised for the 4 field scenario, indicating that a limited number of field moves introduces some flexibility into the system which helps to overcome the impact of short process breakdowns, as was the case for harvester breakdowns. Increasing the number of fields beyond 4 led to a decrease in output due to impact of lost harvesting time due to field moves. When both types of vegetable are harvested the additional complexity inherent in this operation means that the impact of field moves is more limited. No vegetable time increases with the number of fields due to the lost harvesting time due to field moves, although some flattening off after 8 fields is seen.

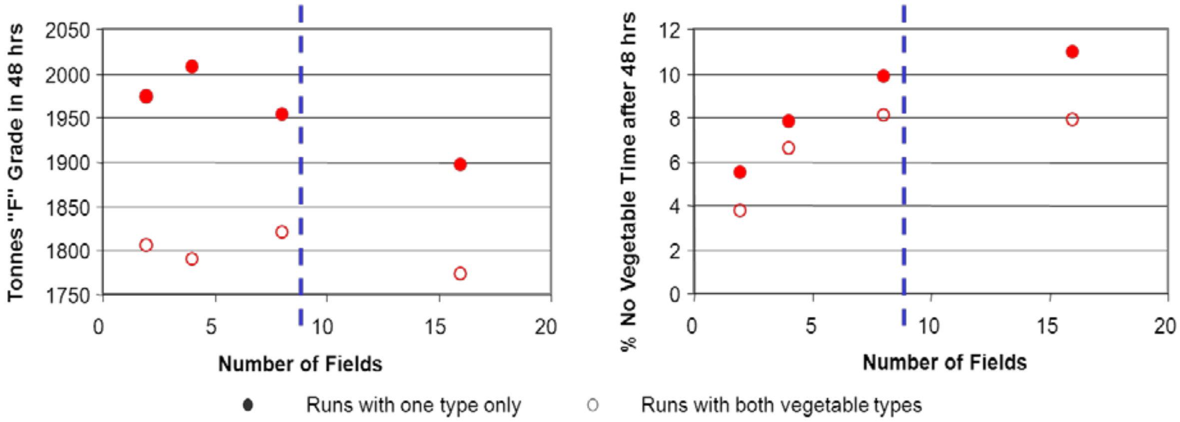


Figure 4 *The impact of increasing the number of fields*

3.4 Quantifying the impact of breakdowns

Figure 3 highlighted that process breakdowns have a significant impact on the output of the model and the factory during the season. Typically the process sees a breakdown rate of 9%, but how the breakdowns are distributed was thought to be important e.g. do lots of short breakdowns or few longer breakdowns have the most significant impact on the process. The model was run with 9% breakdowns, with the breakdowns distributed in different ways, the results are shown in figure 5.

For both cases shown there is a general decrease in the output of “F” grade vegetable produced as the length of the breakdown increases. The difference between results for 10, 20 and 30 minute breakdowns is fairly small. However, there is a significant decrease in output for 60 minute breakdowns, which is by far the worst case. This is due to how the model responds to breakdowns. For breakdowns of 30 minutes or less the model will not stop harvesting, but for longer breakdowns if a large enough tonnage of vegetable is on the road harvesting will stop until the tonnage of vegetable on the road drops to a specified level. Analysis of model results showed that this stoppage time is about 1 hour. So for the case of a 60 minute breakdown harvesting will restart as the line is repaired and the line will be starved of vegetable for at least a further hour, it is therefore not unexpected that 60 minute breakdowns are the worst case. In the 120 min case harvesting will restart at a lower level before the breakdown is fixed hence the impact is lower.

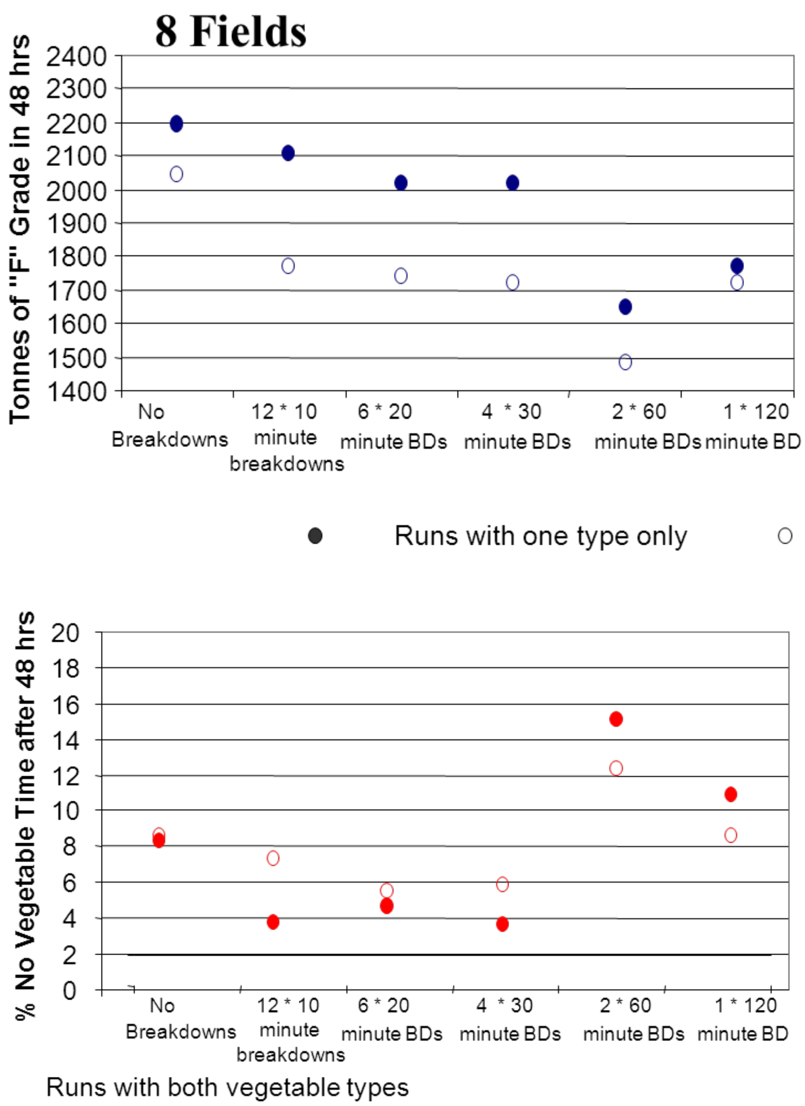


Figure 5 *Quantifying the impact of breakdowns of varying duration*

4 DISCUSSION AND CONCLUSIONS

The model has allowed the team to quantify the impact of a number of fundamental factors on the harvesting operation of this vegetable product. These would be almost impossible to determine in practise, particularly given the time pressures under which the operation functions. This led to a number of recommendations which were adopted in practise.

As the model highlighted and quantified that processing both types of vegetable on the same day has a significant impact on output. The majority of the second type is now processed at the end of the season to minimise the impact. Based on these results the following recommendations are made for running both types of vegetable together:

- Minimise line changeovers.
- Ensure sanitation is dealt with effectively.
- Allocate lines based on field yield as well as available tonnes in a region.
- If there are sufficient tonnes of the second variety run 2 lines together and cover sanitation in this way.

- Try to manage the movement of fields to give a smooth input of the second variety across the day (this will minimise the need for changeovers). The basic premise is that the longer the same combination of lines can be used together the better the outcome.

The major practical impacts of the breakdown results are managerial. It is now clear to the whole team that the impact of the factory breaking down is significant, not only must this be minimised, but good communication between field and factory teams is essential. For example if production can let the agriculture department know when a line will restart, harvesting can be restarted before the line returns, significantly decreasing no vegetables time. Information sharing between production and agriculture will lead to fewer issues of phasing delivery of vegetables with current factory capacity and lead to a fuller factory more of the time. Planned improvements in communication between the field and factory teams should improve this situation. In addition the agriculture department should not stop vining completely but cut targets back for the reduced factory capacity.

The findings of this study are not only relevant to this specific context but extend previous work (e.g. Arjona *et al.*, 2001; Iannoni and Mirabilo, 2006) which considered the harvesting and transportation operations of agricultural production operations; by incorporating a simulation model of the processing operation. This work has confirmed previous findings that ensuring a smooth supply is vital to the success of this type of operation for two reasons, to enable good utilisation of production operations and to ensure that all material is processed within the required time. This work has then gone on to highlight the impact of production stoppages due to sanitation and breakdowns on the overall operation. This is particularly relevant to any operation where the time from harvesting to processing is short and hence the impact of a breakdown is significant. In these cases good communication links between harvesting and processing operation teams is vital to minimise the impacts of the downtime. In addition this work has extended previous work by investigating the complexities of coping with a multi-product harvesting and processing operation. The results have shown that multi-product situations add a significant degree of complexity to the operation which is detrimental to efficiency if it is not carefully managed. The best results are achieved by keeping operations simple. If it is possible to dedicate unique production resources to each product so that they can effectively be managed separately this work has shown that efficiency can be greatly improved.

This work could and should be extended in a number of ways. Further work using the model in its current form should look in more depth at the impact of both processing the second variety and breakdowns. Work should examine the impact of harvesting different proportions of the second variety during an operational day and look at different operational strategies for ameliorating the impact of breakdowns. Another issue which was not possible to deal with during this project was the allocation of the wagons, which deliver vegetable from the field to the factory, the model highlighted that the use of these is not always efficient – different models for the use of wagons should be explored. In the future it would be beneficial to extend the model in two ways - firstly to build models of the downstream operation looking to improve the efficiency of the bulk storage and consumer packaging operations and also to build models of other similar operations of which at least one exists within the case company.

The use of a simulation model of this key harvesting and processing operation has provided real insights into the root causes of some of the critical issues which occur every year in the harvesting season. This fundamental understanding will not only improve decision making in future seasons but has led to a number of recommendations for running the operation which have been embraced by the whole harvesting and processing team. In addition the harvesting team now have a three day model available to them which will give them the opportunity to weigh up different alternative strategies for the coming days during the season itself. The model has provided a low risk environment for improving this business critical operation which should continue into the future.

REFERENCES

Andrieu N, Poix C, Josen E and Duru M (2007). Simulation of forage management strategies considering farm-level land diversity: example of dairy farms in Auvergne. *Computers and Electronics in Agriculture,* **55(1)**: 36-48.

Arjona E, Bueno G and Salazar L (2001). An activity simulation model for the analysis of harvesting and transportation systems of a sugarcane plantation. *Computers and Electronics in Agriculture,* **32(3)**: 247-264.

Bradley D P and Winsauer S A (2004). Solving wood chip transport problems with computer simulation. Available from: <http://www.ncrs.fs.fed.us/pubs/rp/rp_nc138.pdf> [Accessed: October 2013]

Higgins A and Muchow R C (2003). Assessing the potential benefits of alternative chain supply arrangements in the Australian sugar industry. *Agricultural Systems* **76(2)**: 623–638.

Iannoni A P and Morabito R (2006). A discrete simulation analysis of a logistics supply system. *Transportation Research Part E* **42**: 191 – 210

Martin J P, Torrijos A, Del Olmo V G (2002). Timber Logistics in Galicia, Spain. *Fourth International Meeting for Research in Logistics,* Lisbon

Mathew J and Rajendran C (1993). Scheduling of maintenance activities in a sugar industry using simulation *Computers in Industry* **21(3)**:331-334

Neves M F, Zylbersztajn D and Neves E M (1998). The orange juice food chain. In *Proceedings of the Third International Conference on Chain Management in Agribusiness and the Food Industry*. Wageningen Agricultural University, Netherlands, pp. 437–447.

Semenzato R (1995). A simulation study of sugarcane harvesting. *Agricultural Systems* **47(4)**:427-437

AUTHOR BIOGRAPHY

NICKY YATES is a lecturer in logistics and supply chain management in Cranfield School of Management’s Centre for Logistics and Supply Chain Management. She specialises in the teaching and research of the modelling of supply chains, including simulation and network modelling; with a particular interest in supply chains which handle perishable products such as blood and food. Nicky has a BEng and a PhD in Chemical Engineering and prior to joining Cranfield spent 6 years doing process development in the food industry.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

A FRAMEWORK FOR DEVELOPING SIMULATION-BASED SERIOUS GAMES FOR OPERATIONS MANAGEMENT EDUCATION

Dr. Durk-Jouke van der Zee

University of Groningen
P.O. Box 800
9700 AV Groningen
The Netherlands
d.j.van.der.zee@rug.nl

Steffan Sloot MSc

University of Groningen
P.O. Box 800
9700 AV Groningen
The Netherlands
s.sloot@student.rug.nl

ABSTRACT

Allowing operations management students to practice their skills in realistic type of environments may imply great help for them in preparing for their future careers. In many cases such practicing is not possible or desirable in a real-life setting. Many authors show how simulation-based serious games may be successfully used instead. Unfortunately, modelling methodology for developing suchlike games is fragmented along disciplinary lines, i.e., teaching design, serious game design and simulation modelling. To address this gap we propose a framework for developing simulation-based serious games. It structures design activities by acknowledging three aggregation levels, the teaching method, the serious game, and the simulation model. Contributions made by simulation are linked to elementary design decisions. Framework use is illustrated by a case example. The example concerns the development of a simulation-based serious game allowing students to play the role of the operations manager of an emergency department.

Keywords: Simulation, serious games, operations management, education

1 INTRODUCTION

This paper was motivated by the observation that increasing demands for health care due to, among others, an aging population, new medical technologies or treatments, and greater welfare are and will be putting high pressure on health care operations management educators and education. Such pressures are likely to become apparent in terms of large numbers of university students – in answering to a rising number of vacancies in health care operations management – and the need for much course development, as educational support (for example, literature, case studies) is rather scarce. Among others, (new) educational programs should allow students to experience real-life problems to foster their creative skills in problem solving – just like this is the case for other disciplines. Training-on-the-job in terms of company visits or internships may not (always) solve matters here, due to, for example, large numbers of students, and for reasons of patient safety or privacy.

Simulation-based serious games may provide an answer for the students' needs on experiencing realistic type of settings in mastering operations management skills. Note how the adjective "serious" stresses the educational purpose of the game (Crookall, 2010). Many authors report successful use of discrete event simulation for serious gaming in operations management education, see Van der Zee et al. (2012) for examples.

Unfortunately, potential and success of simulation-based serious gaming does not seem to go together with the existence and development of simulation-based modelling methodology for serious game development. Not surprisingly, aforementioned authors proposing simulation-based serious games for operations management education only provide scarce information on methodology applied for game development. Clearly, the observed lack of modelling support for game designers impacts on modelling efficiencies and effectiveness. Moreover, much of the potential of simulation for educational purposes may not be recognized or reaped, simply because a clear course of action linking educational needs to simulation features is missing. In other words, the availability of simulation modelling methodology for serious gaming purposes may not only affect the success of the modelling effort but also the use of simulation-based serious games for educational purposes.

Starting from the observed gap on simulation modelling methodology and previous research on conceptual modelling for simulation-based serious gaming (Van der Zee et al. 2012) we propose a high-level framework for developing simulation-based serious games for operations management education. In proposing the framework we seek to accomplish two aims: (i) to provide guidance to the serious game designer or design team by structuring the development process in a step-wise manner, thereby linking educational needs and simulation modelling efforts, (ii) clarify potential of simulation for serious gaming for educators by considering its contributions for implementing learning activities, and role and functionalities for game operation. The framework is typified as high-level as we try to link each step to an existing “proven” method for the design of teaching methods, serious games, simulation conceptual and coded models respectively, rather than defining a new dedicated method. At the same time the framework shows how aforementioned development steps are linked based on simulation features, i.e., its event based time mechanism, allowance for uncertainties in operations’ timing and outcomes, and associated tools offering libraries of building blocks for efficient system modelling and visualisation.

This paper is structured as follows. In Section 2 we introduce research methodology, underlying set-up and use of the new high-level framework for developing simulation-based serious games for operations management education. In Section 3, the framework is discussed in detail. Use of the framework is illustrated by a case example concerning a simulation-based practical for health care operations education (Section 4). Section 5 discusses contributions made by the framework. Main conclusions are summarized in Section 6.

2 METHODOLOGY

2.1 Design of Simulation-Based Serious Games for Operations Management Education

The new framework is meant to support design and use of simulation-based serious games for operations management education by identifying, structuring and facilitating simulation modelling activities. Relevant issues in operations management education concern systems design (for example, shop floor layout, supply chain design) and their planning and control (for example, releasing, dispatching, planning). The games ‘serious nature’ is related to the students’ needs for practicing their decision making skills in a “realistic environment”. Here students are meant to learn by “using their developing concepts to improve their actions: to put the theory into practice in working to a goal, generating an action to achieve it, and using the feedback to modulate their action or their conception” (Laurillard 2012).

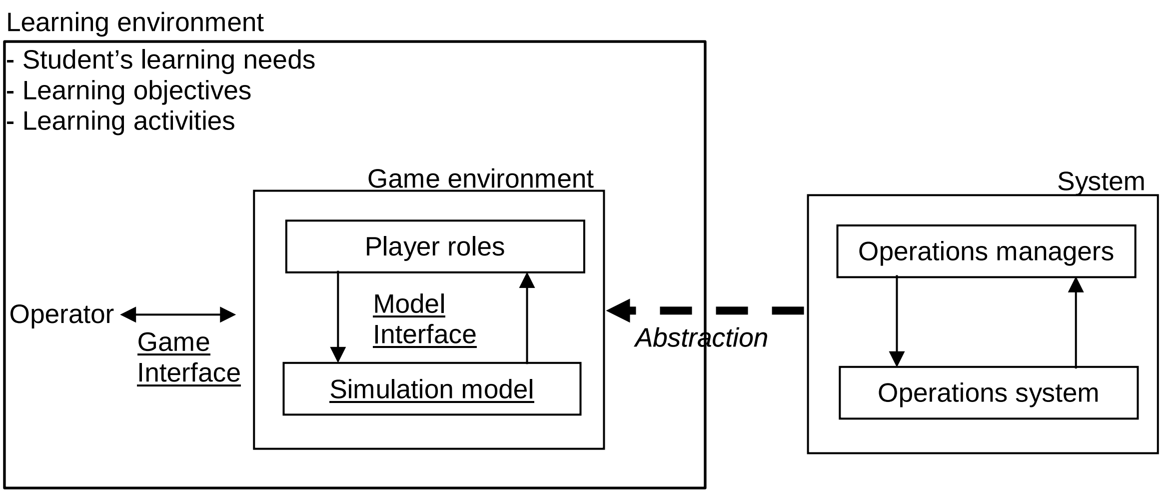


Figure 1 *Game Environment within Learning Environment – Computer Simulation Support (adapted from Van der Zee et al., 2012)*

Framework support is meant to address two perspectives. The perspective of the game designer is addressed by identifying, and linking key steps in serious game design – being initiated by an educator, and being supported by computer simulation. The perspective of the educator is meant to be facilitated by highlighting potential contributions of simulation features in answering to educational needs for students practicing their skills. By addressing these interrelated perspectives we serve both the "customer" (educator) and the "supplier" (game designer) of the game to be developed.

Computer simulation support is assumed to address three game elements within a learning environment as characterized by student's learning needs, learning objectives, and learning activities (Figure 1): (i) a simulation model capturing relevant detail and the status of a referent operations system, (ii) a model interface facilitating players', i.e., students' roles as operations managers, by informing them on model status and performance and implementing their decisions (iii) a game interface enabling the operator to intervene in game set-up (initial settings) and progress (Van der Zee et al., 2012). Together, student's learning needs and (course) learning objectives underpin the choice of learning activities. The simulation-based serious game is meant to support activities that aim to foster learning through practicing, i.e., enabling the student to understand and use the knowledge and skills of a discipline (Laurillard 2012).

2.2 Aggregation Levels in Simulation-Based Serious Game Design

The new framework is meant to guide the process of designing simulation models for game use in an educational environment. Three levels of aggregation are considered relevant for simulation-based serious game design. The highest level is made up by the design of the teaching method, i.e., a learning outcome oriented set of activities to be performed by learners and learning supporters (Derntl et al. 2009). Next levels detail respective activities and their support. Our framework considers two such levels, being specialized towards the design of the educational medium for facilitating learning through practicing (i.e. serious game), and the design of the simulation model for enhancing the gaming experience through computer-based support. Note how respective levels are decoupled by decision making on the choice of learning environment, i.e. gaming vs. a real-life setting, and the need for computer simulation support.

2.3 Towards a Framework for Developing Simulation-Based Serious Games

In the previous section we related simulation-based serious game design to three levels of aggregation, focusing at the design of the teaching method, the serious game, and the computer simulation model respectively. The framework is meant to support and promote use of simulation-based serious games by identifying, structuring and facilitating design activities contributing to efficient and effective use of simulation. To do so, the framework relates each aggregation level (design step) to a series of key-design activities and their associated support. Choice of activities is related to existing "proven" methods for the design of teaching methods, serious games, simulation conceptual and coded models respectively.

3 A FRAMEWORK FOR DEVELOPING SIMULATION-BASED SERIOUS GAMES FOR OPERATIONS MANAGEMENT EDUCATION

3.1 Framework Overview

The new framework distinguishes between three levels of aggregation in simulation-based serious game design. Respective levels focus at the design of the teaching method (I), serious game (II) and the simulation model (III), see Figure 2. Each aggregation level is associated with a series of design activities. For the third level, i.e., the simulation model, we distinguish between three subsequent modelling steps: the simulation conceptual model, the simulation coded model and the simulation use model. The simulation conceptual model is meant to capture the specification for the coded simulation model, in terms of modelling objectives, model contents, inputs and outputs (Robinson 2008b). Next, a simulation tool is used to implement the conceptual model. Finally, the respective coded model may be tailored towards a model addressing a specific educational setting, i.e., simulation use model, by adjusting its inputs and initialization in order to represent alternative game scenarios.

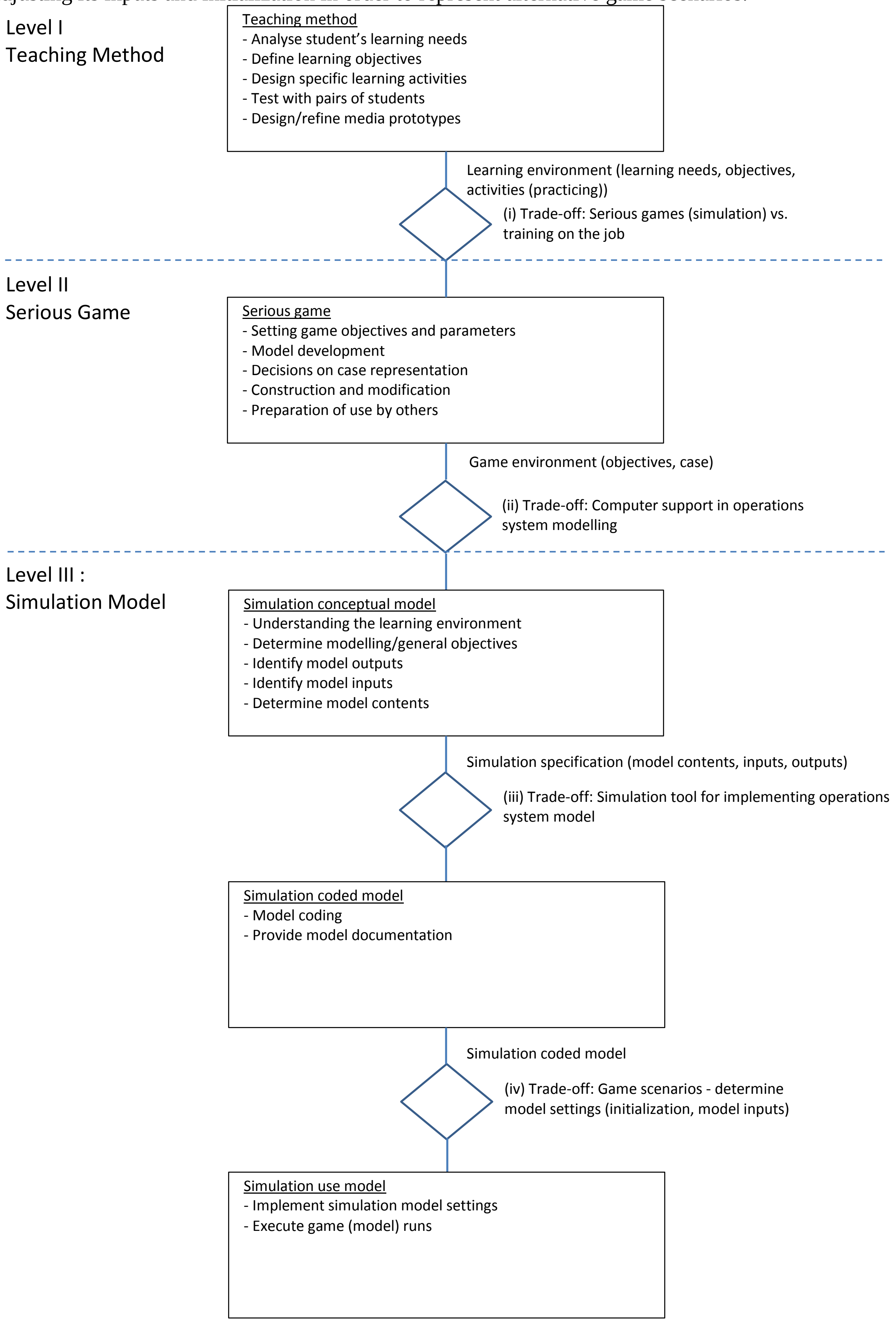


Figure 2 *A Framework for Developing Simulation-based Serious Games for Operations Management*

Aggregation levels are decoupled by decisions on (i) the implementation of the learning environment for practicing student's skills, i.e., trading-off training on the job and gaming, (ii) the use of computer support in abstracting the operations system, (iii) the choice of simulation tool for implementing the simulation conceptual model, and (iv) the choice of parameter settings of the simulation model to tailor it for a specific educational setting.

By identifying and structuring activities in simulation-based serious game design, and their associated support in terms of methods, tools, and good practices (Section 3.2) we attempt to provide guidance to the game designer. The educator is meant to be facilitated by highlighting potential contributions of simulation features in terms of key decisions in game development (Section 3.3).

3.2 Design Activities

3.2.1 Teaching Method

For the design of the teaching method we rely on the work of Laurillard (2002, 2012), given her focus on the good use of digital technologies for learning and teaching in higher education. Learning objectives and student's learning needs, given his or her current understanding of a subject, are the starting points for the design of a teaching method. Choice and design of learning activities for composing the teaching method is considered a largely creative process. For our framework we focus on those activities that assume student's learning through practicing. Essentially, practicing may be related to a real-life setting – as in training on the job (for example, internships or company projects) – or an emulation of such a setting – as in serious games. In turn, serious games may be supported by digital technologies like simulation, the web and immersive environments. Developmental testing is meant to judge the extent to which the design of the learning activities enables students to achieve the intended objectives. Testing teaching methods with students is considered a prerequisite.

3.2.2 Serious Game

The game design process is characterized according to Greenblat (1988). It builds on earlier work with Duke (1981). It serves as a reference model for simulation and gaming – being much cited by simulation game designers, see Van der Zee et al. (2012) for references. Greenblat distinguishes between five stages in game design. The first stage concerns the setting of game objectives and parameters. It addresses game subject matter, identifies players and game operators, context of use (for example, workshop or course), and resources available for game development and operation. In the second stage, i.e., model development, essential characteristics of an imaginary or real system (compare operations system in Figure 1) are captured, including relevant actors (compare operations managers in Figure 1). The third stage complements stage II by considering representational style and form for model elements, including detail for elements, time frame for game operation, ordering of game activities, and player (student) interaction. Final stages consider game construction and preparation. Game construction includes decision making on computer use in game design.

3.2.3 Simulation Model

Three modelling steps are considered for designing the simulation model, i.e., developing the simulation conceptual model, the simulation coded model and the simulation use model. Here we focus on the simulation conceptual model, having presumably the largest impact on simulation model (game) set-up and representation. The nature of design activities concerning the coded model tend to depend strongly on the choice of simulation tool, conforming, for example, to the object oriented paradigm or procedural programming principles, being parameter driven only or allowing for user-defined building blocks, being visual interactive or not etc. Typically, the design of this simulation use model will only require little efforts of the designer as only model parameters have to be adjusted.

For characterizing simulation conceptual modelling activities we build on the modelling framework by Van der Zee et al. (2012). Essentially, the framework results from extensions and modifications of the modelling framework proposed by Robinson (2008). The latter framework is meant as guide for the analyst in developing a simulation model for the purpose of operations systems analysis. Alternatively, the new framework by Van der Zee et al. focuses at simulation use for serious gaming purposes. It considers five activities in conceptual model specification. The initial activity, i.e., understanding the learning environment, links efforts on simulation modelling to outcomes for serious game design (first three activities mentioned). Modelling objectives are meant to express players' (students') achievements in mastering their decision making skills. General project objectives consider requirements on project resources and model nature and use, with respect to visualization, player interaction, responsiveness, and model/component re-use. Specification of model contents is closely linked to the model development activity for defining the serious game. Model inputs, may be employed to create alternative game scenarios, whereas model outputs indicate player's skills in dealing with such scenarios.

3.3 Key Decisions in Simulation-Based Serious Game Design

Important simulation characteristics are its capability of accurately representing operations systems, including uncertainties in operations durations and outcomes, its efficient time mechanism – allowing to "skip" time between events, i.e. moments system status changes, and availability of computer support. Below we relate these characteristics to decisions in simulation-based serious game design.

3.3.1 Emulate Operations Systems?

Operations management education prepares students for future careers in business. How to gain experience before entering business? A game environment is a suitable environment for doing so. Educators may favour games instead of training on the job as in internships and (thesis) projects for reasons of *visibility, reproducibility, safety, economy, and system availability* (Raser 1969). Note how such game qualities meet usual course characteristics, being focused in-depth on a limited set of issues concerning design of operations systems and/or their planning and control, for a large(r) number of students, within a limited period of time. Clearly, in most cases, no real-life operations' system will be available to meet suchlike demands. Remark how student experiences from game use may make their practicing in real-life settings – as in training on the job – more successful as students benefit from prior experiences.

3.3.2 Computer Support?

Computers may be used for various purposes in game operation (Greenblat, 1988; Thavikulwat, 2004; Laurillard, 2012). They may facilitate players by providing access to means for calculation (spread sheet, calculator etc.), and databases. Player communication and interaction may be (partly) relying on computers. Making players familiar with computers or particular software applications may be another objective. Advanced tutoring systems offering personal tuition for players during game operation may supplement teacher activities. Aforementioned uses assume computers to be an aid to the players. Computer models may also be an intrinsic part of the game, representing some complex system that involves activities not easily replaced by those of game operators and/or players. Operations systems may often be classified as suchlike systems.

Current computer simulation support is especially focussing at representing operations systems. Means for (multi) player interaction with the simulation model are usually weakly supported (Van der Zee and Slomp 2009). Typically, player and operator interfacing facilities are no standard features for most simulation tools – being focused at model demonstration rather than player interaction. Links of simulation tools to intelligent tutoring systems seem to be open for further exploration.

3.3.3 Choice of Simulation Tool?

Many simulation tools are available, see Swain (2011) for an overview. Game use, however, stresses some specific demands on tool features (Van der Zee and Slomp, 2009). We especially mention (i) facilities for realistic visualization of operations systems and their dynamics, and (ii) means for developing player/operator interfaces. More advanced features may be required in case the simulation model has to accommodate a multi-player environments, integrate the notion of an intelligent tutoring system, or integrate player use of decision making tools like, for example, spread sheets and databases (Greenblat, 1988; Laurillard, 2012). Next to aforementioned features, common criteria for simulation software selection apply, see, for example (Law 2008), as well as dedicated criteria resulting from simulation conceptual modelling (general project objectives).

3.3.4 Choice of Scenario?

From a coding perspective the choice of game scenario may often be considered a trivial one, boiling down to parameter setting for model inputs and/or model initialization. The educator’s job may be more difficult here, as he/she has to match students’ learning needs to model settings.

4 CASE EXAMPLE – DEVELOPING A SIMULATION-BASED PRACTICAL FOR HEALTH CARE OPERATIONS EDUCATION

4.1 Background

Recently, the authors participated in the set-up of a practical for a new course on health care operations (Sloot, 2013). Design of the course assumes a two phase approach in student learning on health care operations management. The first phase heavily employs lecturing as a means to acquire knowledge and foster understanding among students on health care operations management. In the second phase case based practicals are used to develop student’s skills in employing their findings from the first phase.

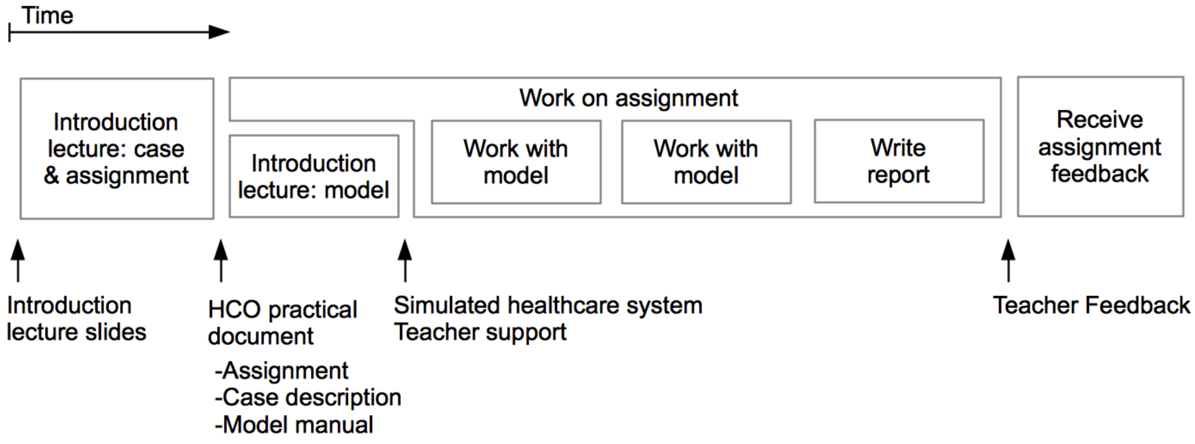


Figure 3 *Practical Set-up* (adapted from Sloot 2013)

4.2 Game Development

4.2.1 Teaching Method

By doing the practical, students need to learn: (i) on the workings of a typical health care system (inputs, processes and outputs) and the influence of variability on system performance, (ii) how logistics decision-making may influence health care system performance, and (iii) how quantitative models may support health care decision making.

The operations system chosen – being exemplary for health care – is the emergency department (ED). The student population addressed by the course suggests learning needs not to be homogeneous. For example, the course serves as an elective for students doing an MSc on, among others, industrial engineering, marketing, human resources management, and operations research. Practical set-up is shown in Figure 3. Core element is an assignment assuming students to do a serious game. In this game they play a role of the manager of an ED, who has to consider and report on possibilities for improving ED logistical performance, given various demand patterns, as defined by patients arrival rate, urgency and needs. Supportive learning activities introduce ED operations and game specifics by a lecture. Feedback on student accomplishments is provided to each student individually during the work on their assignment, and after completing it.

Extensive testing of the practical is done by interviewing students participating in the course. Furthermore, elements of the practical have been tested separately, see Section 4.2.2. and 4.2.3.

4.2.2 Serious Game

Starting from the learning objectives, a case description was defined in a textual format. It describes the operations system for a fictitious ED, and clarifies the role of the ED manager in terms of measures to undertake for improving or guaranteeing ED logistics performance under various scenarios for patient demand. ED set-up and the ED manager's role build on observations of three real life EDs, and literature. Moreover, the case description has been validated by ED managers of aforementioned EDs. Further refinements followed from employing a student panel that checked the case description for consistency and clarity.

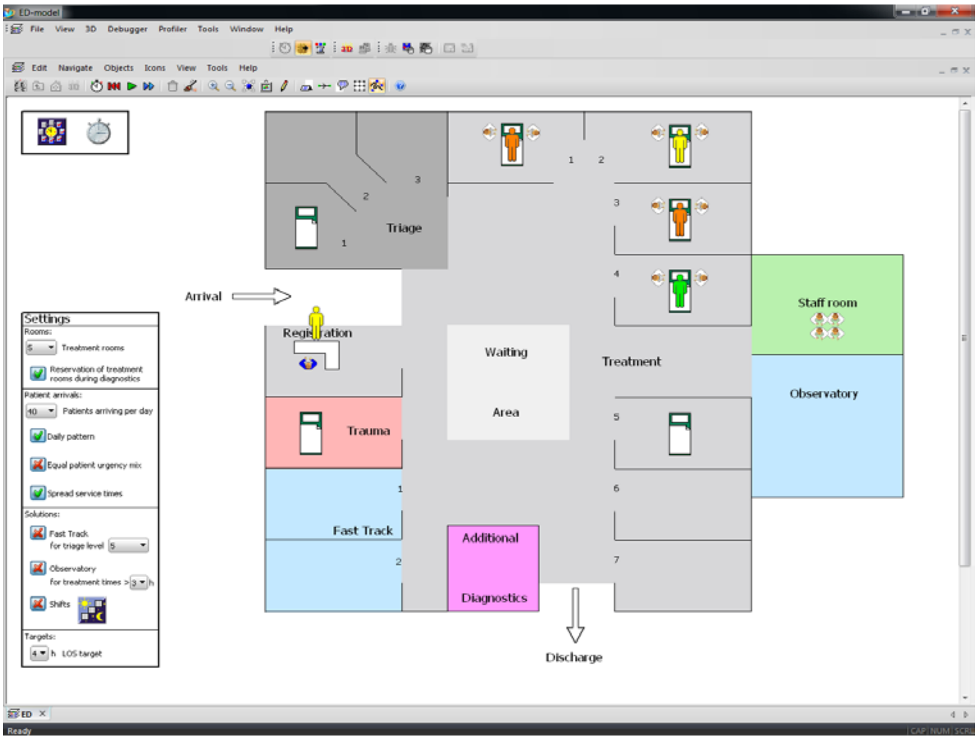


Figure 4 *Coded Simulation Model for Emergency Department*

4.2.3 Simulation Model

The simulation conceptual model was defined in a rather straightforward manner, starting from the case description, and by employing the framework by Van der Zee et al. (2012). Main elements added in the conceptual model, apart from the restructuring of the case description into a specification for model coding, concerned clarifying demands related to model use, i.e., its required features with respect to visualization, operator/player interaction, responsiveness and model component re-use. Furthermore, only preliminary parameter ranges for experimental factors could be defined – to be put to the test when the coded model is available. The conceptual model was validated in an indirect way by letting ED managers and students assess the case description, see 4.2.2.

The coded simulation model is built using Plant Simulation (Siemens, 2013), see Figure 4. It is supplemented by a user manual. Also instructions on its use are given during introductory lectures, see Section 4.2.1. A simulation use model is defined for executing the practical in January 2013. Effectively, this boiled down to restricting the student in his role of an ED manager in the type of measures he/she could take for influencing ED performance (cf. Setting in Figure 4). In this way case complexity and student efforts are reduced.

4.3 Decision Making on Game Design

The choice for a game setting in practical set-up was motivated by several reasons:

- The possibility to adjust case settings for focusing on just those issues addressed in the course.
- Its allowance for player testing of various scenarios and measures within a short time period.
- Issues of patient safety and privacy do not allow student activities within a real-life setting.
- Student numbers are simply too large to handle, even by a mid or large size ED.

Computer support was motivated by (i) the simulation model being an intrinsic part of the game. Only, by providing a computer model of the - complex - ED system, the practical would be do-able within resource and time constraints, and (ii) the idea of making students familiar with simulation tools. Plant Simulation was chosen as a simulation tool, mainly because of its libraries of building blocks for modelling and visualizing of operations systems, and facilities for providing player interaction. Last but not least, it was readily available.

5 DISCUSSION

5.1 The Framework – Being Explicit on Simulation-based Serious Game Design

Typically, methodology for designing simulation-based serious games is fragmented along disciplinary lines, i.e., teaching design, serious gaming, and simulation modelling. The framework proposed in this article seeks to address this gap in modelling methodology, by identifying, structuring, aligning and facilitating design activities. Here choice of design activities is related to well-known methods for the respective fields being addressed. Knowing what activities to address, and how to align them is assumed to contribute to effectiveness of simulation-based serious games in terms of learning objectives being met and modelling efficiencies. Moreover, making such knowledge explicit and available may help to exploit the great potential of simulation for educational use, instead of making it a luxury item, whose existence depends on the accidental blending of the required design skills.

5.2 Scope

Our choice of domain suggests a wide field of application for simulation-based serious games. It may be worthwhile to tailor the framework to specific classes of operations systems. As far as the learning environment is concerned our framework focuses at educational use rather than training. Extensions of the framework to accommodate training should account for a need to tailor game set-up and use to (in)company preferences, as reflected in player (trainees) backgrounds and operations systems detail.

5.3 Guidance – Where Education, Gaming and Simulation Meet

Teaching design, serious game design, and simulation modelling all are highly creative activities, often considered to be more of an art than a science. Usually, the designer’s experience is a prime input for such creativity. By proposing the framework we aim to structure development of simulation-based serious games by acknowledging various aggregation levels in their design, corresponding to the activities of the teacher being responsible for designing a course, the game designer seeking to facilitate learning activities that suggest student's practicing by creating a game environment, and the simulation modeller offering computer support for such an environment. Moreover, key decisions in simulation-based game design are identified that link good use of simulation to educational needs.

6 CONCLUDING REMARKS

In this article we proposed a framework for designing simulation-based serious games, starting from the observed potential and need of suchlike games for operations management education. The framework distinguishes between aggregation levels, addressing the design of the teaching method, the serious game, and the simulation model respectively. Each aggregation level is related to a series of design activities, being common to teaching design, serious game design, and simulation modelling respectively. Key decisions in simulation-based game design are identified that link good use of simulation to educational needs. Framework use is illustrated by a case example concerning the design of a computer practical for a health care operations course, entailing a game in which students take a role as a manager of an emergency department.

Future research is directed towards the further refinement of the framework, especially aligning design activities, thereby building on more case examples. Apart from its role for serious game design the framework shows how the potential of simulation may be further exploited, by explicitly linking its use and strong points to observed needs.

REFERENCES

Crookall D (2010). Serious Games, Debriefing, and Simulation/Gaming as a Discipline. *Simulation & Gaming* **41(6)**: 898-921.

Derntl M, Neumann S and Oberhuemer P (2009) Report on the Standardized Description of Instructional Models. ECP 2007 EDU 417007. Available via http://dspace.ou.nl/handle/1820/2057 [accessed October 9, 2013].

Greenblat C S and Duke R D (1981). *Principles and practices of gaming simulation*. Sage Publications: London.

Greenblat C S (1988). *Designing games and simulations – An illustrated handbook*. Sage Publications: London.

Laurillard D (2002). *Rethinking University Teaching – A Framework for the Effective Use of Learning Technologies*. 2nd Edition. Routledge: New York.

Laurillard D (2012). *Teaching as a Design Science – Building Pedagogical Patterns for Learning and Technology*. Routledge: New York.

Raser J R (1969). *Simulation and society: An exploration of scientific gaming*. Allyn and Bacon: Boston.

Robinson, S (2008). Conceptual modelling for simulation Part II: a framework for conceptual modelling. *Journal of the Operational Research Society* **59(3)**: 291-304.

Siemens (2013) Plant Simulation. Siemens PLM 2012. Available via http://www.plm.automation.siemens.com/en_us/products/tecnomatix/plant_design/plant_simulation.shtml [accessed October 9, 2013].

Sloot S (2013) A Simulation-Based Practical for a Health Care Operations Course – Design & Implementation. MSc Thesis, University of Groningen. The Netherlands.

Swain J (2011) Simulation software survey. *OR/MS Today* **38(5)**. Available via http://www.orms-today.org/surveys/Simulation/Simulation.html [accessed October 9, 2013].

Thavikulwat P (2004). The architecture of computerized business gaming simulations. *Simulation & Gaming* **35(2)**: 242-269.

Van der Zee DJ and Slomp J (2009). Simulation as tool for gaming and training in operations management – a case study. *Journal of Simulation* **3(1)**: 17-28.

Van der Zee DJ, Holkenborg H B M and Robinson S (2012). Conceptual modeling for Simulation-based Serious Gaming. *Decision Support Systems* **54(1)**: 33-45.

AUTHOR BIOGRAPHIES

DURK-JOUKE VAN DER ZEE is associate professor of Operations at the Faculty of Economics and Business, University of Groningen, The Netherlands. His research interests include simulation methodology and applications, simulation & serious gaming, manufacturing planning & control, and health care logistic systems engineering.

STEFFAN SLOOT is a junior researcher at the Faculty of Economics and Business, University of Groningen, The Netherlands. His research interests include simulation methodology and applications in health care logistics.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

"JAAMSIM" DESCRIBED IN THREE SIMPLE EXAMPLES

Dr. D. H. King

Ausenco
855 Homer Street
Vancouver, BC, V6B 2W2, CANADA
harry.king@ausenco.com

Harvey S. Harrison

Ausenco
855 Homer Street
Vancouver, BC, V6B 2W2, CANADA
harvey.harrison@ausenco.com

Matt Chudleigh

Ausenco
855 Homer Street
Vancouver, BC, V6B 2W2, CANADA
matt.chudleigh@ausenco.com

ABSTRACT

This paper introduces JaamSim, a free, open source simulation software package that includes a modern graphical user interface, drag-and-drop model building, and 3D graphics. Three simple examples are used to demonstrate the features of the software.

Keywords: JaamSim, Simulation, Discrete Event, Software, Open Source, Java, 3D, GUI

1 INTRODUCTION

This article describes the capabilities of "JaamSim", a free, open source simulation program developed by Ausenco for the benefit of the simulation community. To the best of our knowledge, JaamSim is the first open-source simulation package that provides a modern graphical user interface comparable to commercial off-the-shelf simulation software. We use three simple models to demonstrate some of JaamSim's features.

JaamSim is written in the Java programming language and can be downloaded from the JaamSim website (www.JaamSim.com, accessed Feb. 12, 2014). The executable, source code, user manual, programming manual, and the examples shown in this article are available from the website.

The key feature that makes JaamSim different from commercial off-the-shelf simulation software is that it allows users to create their own palettes of objects for new applications in a standard programming language instead of a proprietary language that is unique to a single software vendor. Programming is done in standard Java using modern development tools such as Eclipse. New objects programmed by the user automatically have 3D graphics, can be dragged-and-dropped, have their inputs editable through the Input Editor, and their outputs displayed in the Output Viewer.

In addition to being extendable, JaamSim is scalable to arbitrarily complex models and provides high quality 3D graphics suitable for artistic animations, training simulations, and serious games.

JaamSim was first developed in 2002 as part of a proprietary simulation application for the oil & gas and mining industries (King and Harrison, 2010). Although it was released as open source software in late 2011, we have waited until it had matured somewhat before introducing it to the simulation community in a recent article (King and Harrison, 2013). A comparison of JaamSim to other simulation offerings in Java can be found in this latter article.

Finally, the authors would like to make it completely clear that we are offering JaamSim as a genuine open source project for the benefit of the simulation community and that we have no plans to introduce a paid "premium" version of JaamSim.

2 JAAMSIM GRAPHICAL USER INTERFACE

JaamSim's graphical user interface (GUI) is a distinguishing feature of the software, so we will begin by providing an overview of its major components. Figure 1 shows the six main windows that make up the GUI.

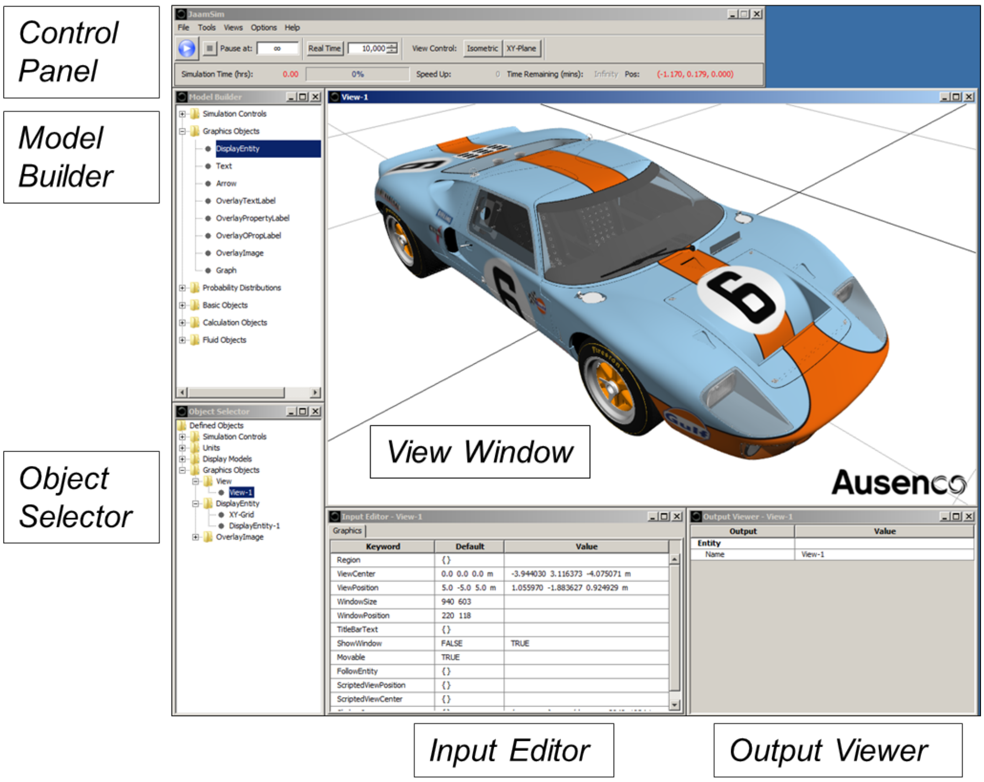


Figure 1 *JaamSim User Interface*

The windows shown in Figure 1 are:

- Control Panel – controls the execution of models and provides access to the other GUI components
- View Window – one or more windows showing 3D views of the model universe
- Model Builder – tool for dragging and dropping model components
- Object Selector – provides access to each object in the model
- Input Editor – displays the inputs for the selected object and permits editing
- Output Viewer – displays the outputs for the selected object

3 SERVER AND QUEUE EXAMPLE

The server and queue are two of the standard building blocks provided with every simulation package. The first example for most simulation software is a barber shop, a bank teller, or some other equivalent server/queue system. In our version of this example, the server/queue system has been expressed in the abstract form shown in Figure 2.

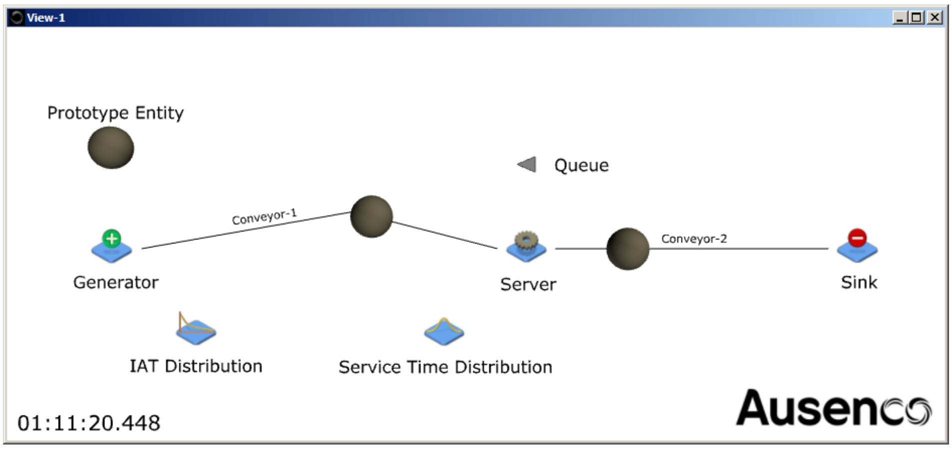


Figure 2 *Server and Queue Example*

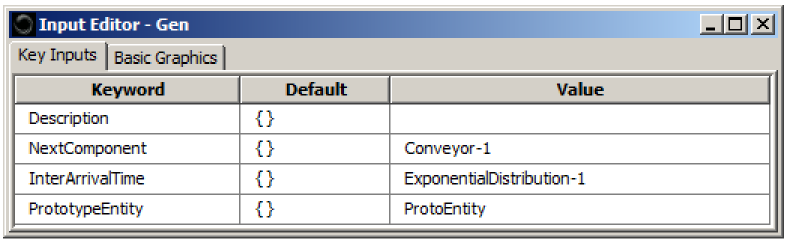
Spherical entities are created by the generator and sent via a conveyor to the server. The server processes the entities and sends them via a second conveyor to the sink where they are destroyed. The entities created by the generator are copies of the prototype entity and are generated at a rate determined by the inter-arrival time (IAT) distribution. The service time for the server is determined by the service time distribution.

The server and queue model provides a good example with which to illustrate some of JaamSim's basic features. The following subsections describe the agent-based structure of JaamSim, its input and output mechanisms, its provision for physical units, and its display of smooth animations.

3.1 Agent-Based Structure

Each of the components in the server and queue model is an independent "agent" that communicates with the other components as equals. The model has the form of a process flow diagram, but this structure was created by the way the various agents were connected to one-another; it is not fundamental to the software. Unlike many other types of simulation software, JaamSim is able to model systems that cannot be expressed as a process flow diagram.

The agent-based logic for JaamSim can be illustrated by examining the inputs for the generator object. Figure 3 shows the entries in the Input Editor for the generator (named "Gen" in the example).


Input Editor - Gen

Key Inputs | Basic Graphics

Keyword	Default	Value
Description	{}	
NextComponent	{}	Conveyor-1
InterArrivalTime	{}	ExponentialDistribution-1
PrototypeEntity	{}	ProtoEntity

Figure 3 *Inputs for the Generator*

The inputs for the keywords NextComponent, InterArrivalTime, and PrototypeEntity completely determine the behavior of the generator:

- NextComponent – the destination for the generated entities
- InterArrivalTime – the time between generated entities
- PrototypeEntity – the entity to be copied

The generator, prototype entity, conveyor, and probability distribution all interact with one another to provide the entity generation function in the model. There is no additional structure that makes any one of these entities special in some way.

3.2 Model Inputs

A JaamSim model is completely data-driven by the inputs to its objects' keywords. The appearance of the model and the positions of the components in the window have no effect on the simulation results.

Model inputs can be entered using the Input Editor or by modifying the input file directly using a text editor. For simple models such as the ones described in this article, the Input Editor is the most appropriate method.

One of the features of the Input Editor is that definition of a keyword can be obtained by hovering the mouse over the keyword, which generates a pop-up. An example of this pop-up is given in Figure 4 for the InterArrivalTime keyword.

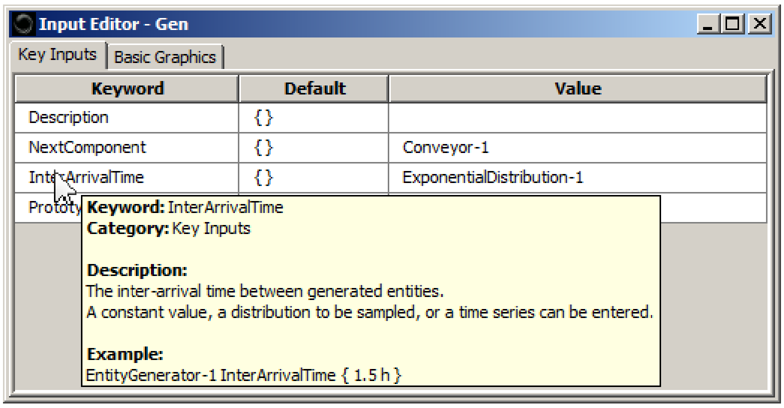


Figure 4 *Keyword Definition Pop-Up*

The text for a keyword pop-up appears next to that keyword in the program code, making the software self-documenting and easier to maintain.

Many keywords in JaamSim, such as InterArrivalTime, can accept a constant number, a probability distribution, or a time series. For example, if a constant inter-arrival time of 100 seconds is desired, the entry in the Value column could be changed to 100 s, as shown in Figure 5.

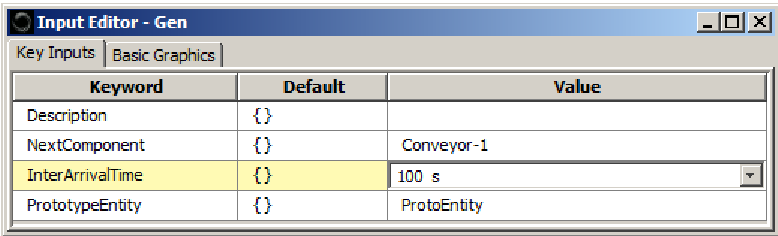


Figure 5 *Input of a Constant Value instead of a Distribution*

3.3 Physical Units

When the entry for InterArrivalTime was changed to 100 s in Figure 5, it was mandatory to include the unit "s". All numerical inputs in JaamSim are identified by the type of unit that is required. For the case of the InterArrivalTime input, a time unit is required. Any valid time unit could have been used, for example, an input of 1000 ms is equivalent to 1 s. All internal calculations in JaamSim are done in SI units (meters, seconds, kilograms, etc.). An input in a non-SI unit is converted to SI when the input is first entered.

The enforcement of units for model inputs means that it is necessary to specify the units for any object that returns a number, such as a probability distribution. The inputs for the IAT distribution (named "ExponentialDistribution-1" in the example) shown in Figure 6 illustrate this requirement.

Input Editor - ExponentialDistribution-1

Key Inputs | Basic Graphics

Keyword	Default	Value
Description	{}	
UnitType	{}	TimeUnit
RandomSeed	1	1
MinValue	0.0 s	
MaxValue	Infinity s	
Mean	1.0 s	100 s

Figure 6 *Specifying the Unit Type*

For an object such as a probability distribution that returns a number, the keyword "UnitType" is used to specify the units for the distribution. In this example, the unit type "TimeUnit" was chosen because this distribution was input for the InterArrivalTime keyword for the generator, which expects to receive a number with the units of time. Many other unit types are pre-defined in JaamSim: DistanceUnit, MassUnit, SpeedUnit, etc. The unit type "DimensionlessUnit" is used to indicate an input that is a pure number. A drop-down menu for the UnitType keyword allows the user to choose from the available unit types.

The unit type chosen for the distribution also determines the unit type for many of the other keywords. In this example, the keywords MinValue, MaxValue, and Mean were all set to expect a time unit once the UnitType keyword was set to TimeUnit. For this reason, the input for the UnitType keyword must entered before any other inputs.

3.4 Model Outputs

Model outputs can be examined through the Output Viewer. Figure 7 shows the outputs for the queue object in the server and queue model.

Output Viewer - Que

Output	Value
Entity	
Description	
Name	Que
DisplayEntity	
Orientation	0.0 0.0 0.0 rad
Alignment	0.0 0.0 0.0
Position	0.493151 2.431863 0.0 m
Size	0.431764 0.365344 0.0 m
Queue	
QueueLength	0.00000
NumberAdded	21.0000
NumberRemoved	21.0000
QueueLengthAverage	0.304201
QueueLengthMinimum	0.00000
QueueLengthMaximum	1.00000
QueueLengthDistribution	{.6957986856223171, .30420131437768294}
QueueLengthStandardDeviation	0.460068

Figure 7 *Output Viewer*

Outputs are grouped by subclasses in the viewer. In this case, the class hierarchy is Entity → DisplayEntity → Queue. An entity's outputs form a complete description of its present state and the statistics collected to the present time. An output's definition can be displayed by hovering the mouse over the output name in the Output Viewer, in the same way as for input keywords in the Input Editor.

Outputs generated by JaamSim are valid at any time, not just at certain event times. To accomplish this, each output is freshly calculated from entity's internal state every time it is required. This arrangement is not as great a computational burden as it might appear, since only a small fraction of the available outputs are required at any one time. An output is calculated only when it is displayed in the Output Viewer, displayed in a Text object, or used for animation.

3.5 Discrete versus Continuous Time

The defining characteristic of a discrete-event simulation is the use of next-event time advance, that is, time is advanced discontinuously from one event to the next. This logic is much more efficient and flexible than advancing time by a fixed increment, but is not suitable when animation is required. An animation must show smooth continuous motion for it to be understandable by the viewer.

For the case of the server and queue example, each sphere moves smoothly along the conveyor even though there are only two events per sphere: one at the start of the trip and one at the end. The motion is smooth no matter how slowly the simulation is executed. Furthermore, the model can be paused at any time, not just at the event times.

This behavior is accomplished by introducing a separate, continuous simulation time that is calculated by interpolating between event times using the computer's system clock. To avoid inconsistent simulation results, this continuous simulation time is only used for animation. This approach allows JaamSim to update the animation each time the scene is rendered, not merely at selected event times.

4 HARMONIC OSCILLATOR EXAMPLE

The second example shows how JaamSim can be used to model continuous variables such as position and velocity. In this example, a damped harmonic oscillator is created by combining two integrators and a weighted sum, as shown in Figure 8.

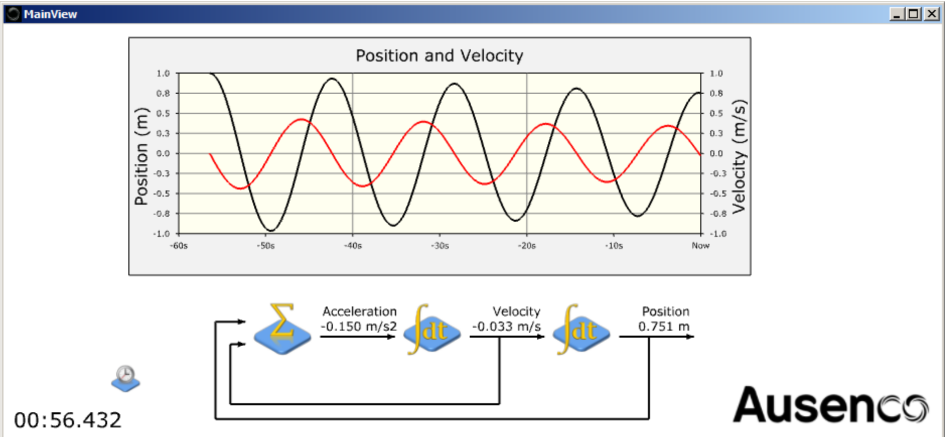


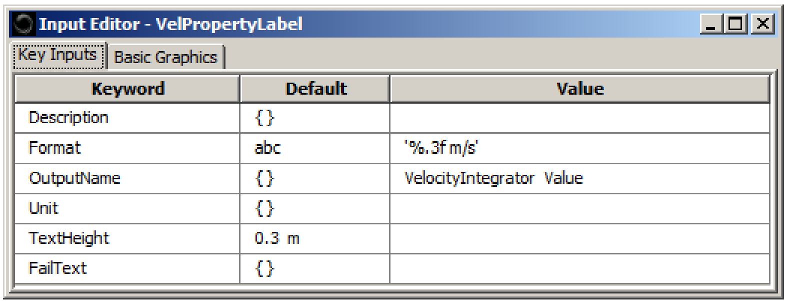
Figure 8 *Harmonic Oscillator Example*

Calculations in the model are updated on a fixed cycle generated by the timer object (the clock-shaped object in the bottom left of the screen). The arrows connecting the various objects are used only to enhance the display. They play no role in the model's logic.

The harmonic oscillator model demonstrates how model outputs can be displayed and graphed.

4.1 Displaying a Model Output

The text displays showing the values of the acceleration, velocity, and position are generated by individual "Text" objects that access the outputs for the weighted sum and the two integrators. Figure 9 shows the inputs for the velocity display (generated by the Text object named "VelPropertyLabel").



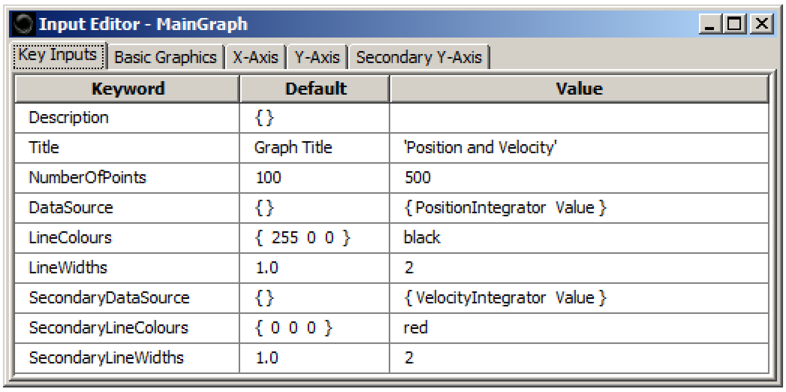
Keyword	Default	Value
Description	{}	
Format	abc	'%.3f m/s'
OutputName	{}	VelocityIntegrator Value
Unit	{}	
TextHeight	0.3 m	
FailText	{}	

Figure 9 *Displaying a Model Output*

The keyword "OutputName" identifies the output quantity to display. The entry { VelocityIntegrator Value } indicates the output named "Value" for the object named "VelocityIntegrator". For this object, the Value output is the present value for the integration. The "Format" keyword determines how the number will be displayed. The input `'%.3f m/s'` is a standard Java format string indicating a floating point number with three decimal places followed by the text `" m/s"`.

4.2 Graphing a Model Output

A similar structure is used to graph a model output. Figure 10 shows the inputs for the "Graph" object name "MainGraph".



Keyword	Default	Value
Description	{}	
Title	Graph Title	'Position and Velocity'
NumberOfPoints	100	500
DataSource	{}	{ PositionIntegrator Value }
LineColours	{ 255 0 0 }	black
LineWidths	1.0	2
SecondaryDataSource	{}	{ VelocityIntegrator Value }
SecondaryLineColours	{ 0 0 0 }	red
SecondaryLineWidths	1.0	2

Figure 10 *Graphing a Model Output*

For this case, the output Value for PositionIntegrator is graphed on the primary y-axis while the output Value for VelocityIntegrator is graphed on the secondary y-axis. These object-output pairs are enclosed by braces in the Input Editor because it is possible to define multiple lines for each axis by entering a list of object-output pairs, with each pair enclosed by braces.

5 FERRARI DAYTONA EXAMPLE

The first two examples have shown models with simple 2D graphics and in many cases, this level of graphics is sufficient for a simulation study. However, it is our ambition to generate high-quality 3D graphics that can be used for animations, training simulators, or serious games. To this end, we have programmed an entirely new 3D rendering system in Java using the JOGL 2 implementation of the OpenGL graphics library (http://jogamp.org/ accessed Nov. 14, 2013).

This example shows a 3D model for a Ferrari Daytona sports car downloaded from the Trimble 3D Warehouse in Collada format (DAE). This is a good quality model consisting of about 50,000 triangles, which is simple enough to be displayed on a typical laptop computer.

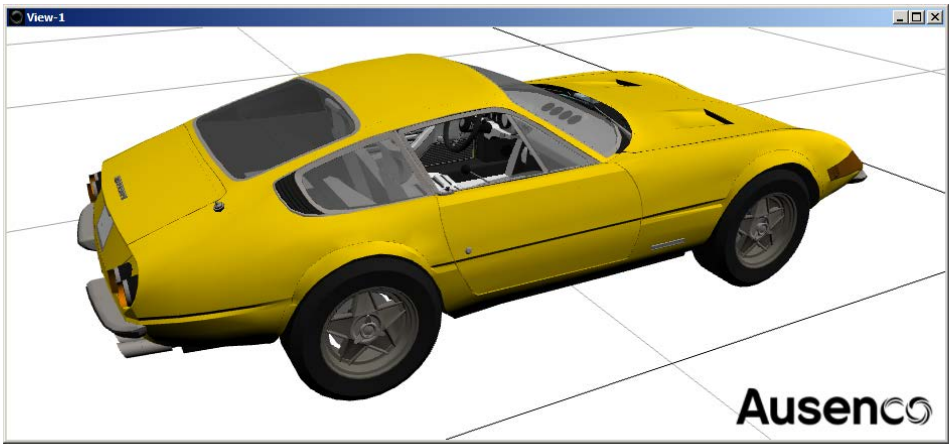


Figure 11 *Ferrari Daytona Example*

5.1 3D Data Formats

JaamSim can import static 3D data in the two most commonly used open formats: Collada (DAE) and Wavefront (OBJ). Two other formats were developed specially for JaamSim: an ASCII format (JSM) and a binary format (JSB). The JSM format is used to import rigged 3D graphics suitable for animation. A plug-in for the Blender graphics program (www.blender.org, accessed Feb. 12, 2014) has been created to export JSM files. The binary format was created to speed up the loading of large graphics files. In DAE or OBJ format, a 3D asset with 5 million triangles can require approximately 1 minute to load. In JSB format, the same 3D asset can be loaded in approximately 10 seconds.

5.2 Sharing of 3D Graphics

In this example, the car is a single "DisplayEntity" whose graphical display has been set to this 3D model. The DisplayEntity class is simplest model element that includes a 3D graphical representation. All the entities we have shown in the previous examples are sub-classes of DisplayEntity.

To allow the same 3D graphics to be used for multiple objects, it is necessary to store the graphic data in a separate object called the "DisplayModel". Figure 12 shows the inputs for the Ferrari Daytona object (named "DisplayEntity-1") with the keyword DisplayModel set to "Ferrari_365_GTB_4_Daytona-1", which was the name of the file downloaded from the 3D warehouse.

Input Editor - DisplayEntity-1

Key Inputs | Basic Graphics

Keyword	Default	Value
Position	0.0 0.0 0.0 m	-0.386940 -1.088423 0.000000 m
Alignment	0.0 0.0 0.0	0.0 0.0 -0.5
Size	1.0 1.0 1.0 m	0.55328 1.35975 0.39534 m
Orientation	0.0 0.0 0.0 rad	
Region	{ }	
RelativeEntity	{ }	
DisplayModel	Cube	Ferrari_365_GTB_4_Daytona-1
Active	TRUE	
Show	TRUE	
Movable	TRUE	
ToolTip	TRUE	

Figure 12 *DisplayModel for the Ferrari Daytona*

To illustrate how one DisplayModel can be shared between multiple objects, Figure 13 shows the server and queue model with the prototype entity set to the Ferrari Daytona instead of a sphere.

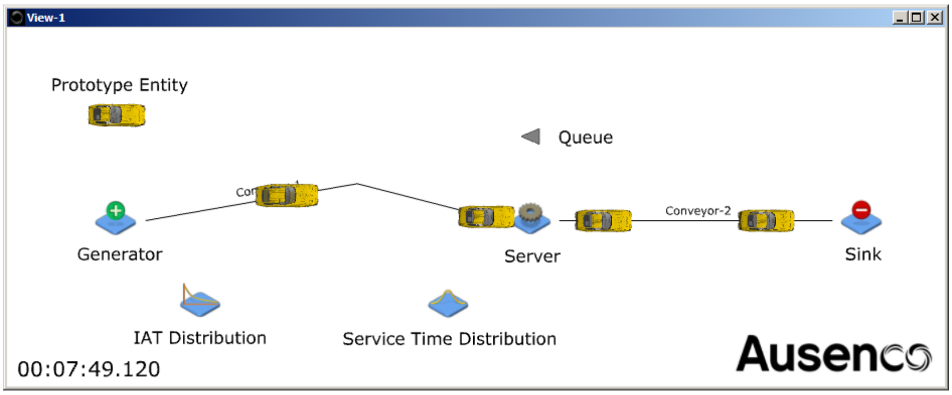


Figure 13 *Multiple Copies of the Ferrari Daytona*

Now, the generator creates copies of the Ferrari Daytona to move through the system. The same DisplayModel is used for each of these copies. Although Figure 13 appears to be a 2D image, it is actually a full 3D scene, as are all graphics generated by JaamSim. The 3D nature of the scene is demonstrated by Figure 14, which shows an oblique view of the same model.

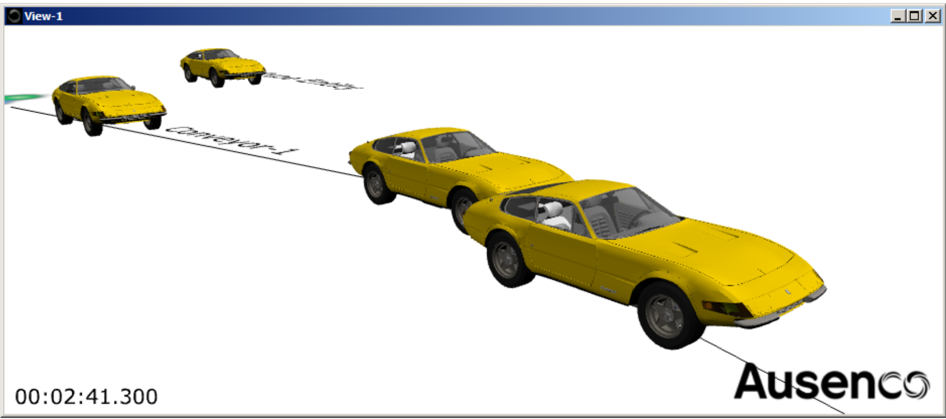


Figure 14 *Oblique View of the Model*

5.3 Animation

It is one thing to import 3D graphics into a simulation, it is quite another to animate them. This is complex topic and we are still in the early days of providing animation capability in JaamSim. The main idea is to link selected outputs for an object to "actions" encoded in the JSM format.

A future article will describe the animation system in more detail.

6 CONCLUSIONS

This article has used three examples to describe some of the features of JaamSim, our free, open source simulation software. At its present state of development, JaamSim offers an excellent user interface, modeling framework, and 3D graphics, but lacks many of the basic objects found in commercial off-the-shelf software.

JaamSim would be a good choice for anyone wanting to developing leading edge models and who is willing to program the objects required for the application in question. If your application requires you to write hundreds of lines of code in the proprietary language provided with your commercial simulation software, you would be far better off to use JaamSim and do your programming in Java.

REFERENCES

King D H and Harrison H S (2013). Open-source simulation software "JaamSim". *Proceedings of the 2013 Winter Simulation Conference*. IEEE: USA, pp 2163-2171.

King D H and Harrison H S (2010). Discrete-event simulation in Java: a practitioner's experience. *Proceedings of the 2010 Conference on Grand Challenges in Modeling & Simulation.* Society for Modeling & Simulation International: USA, pp 436-441.

AUTHOR BIOGRAPHIES

HARRY KING is the Manager of the Simulation Department at Ausenco. Ausenco is a full-service engineering company with a staff of approximately 3,500 people and head office in Brisbane, Australia. The Simulation department is based in Vancouver, Canada and has a staff of 19 professionals. Dr. King holds a Ph.D. in Theoretical Physics from the University of Texas at Austin and has devoted his career to simulation modelling since 1979.

HARVEY HARRISON is the Assistant Manager, Software, for the Simulation Department at Ausenco.
He is the lead programmer for the development of Ausenco's discrete-event simulation software and has experience in many types of modeling including transportation demand models, population change models, vehicle micro-simulation and toll revenue models. He has an interest in systems programming and has many contributions to the Linux kernel.

MATT CHUDLEIGH is a Software Engineer in the Simulation Department at Ausenco. He is responsible for the 3D graphics engine in JaamSim as well as other core systems. During his career he has work in a variety of different fields including biotech instrument design and commercial video game development and has worked as a software engineer and embedded systems engineer. His technical interests include computer graphics and programing language theory.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

EXTENDING AN OPEN-SOURCE DISCRETE EVENT SIMULATION PLATFORM WITH NEW OBJECTS

Georgios Dagkakis

University of Limerick
Enterprise Research Centre
Limerick, Ireland
georgios.dagkakis@ul.ie

Ioannis Papagiannopoulos

University of Limerick
Enterprise Research Centre
Limerick, Ireland
ioannis.papagiannopoulos@ul.ie

Cathal Heavey

University of Limerick
Enterprise Research Centre
Limerick, Ireland
cathal.heavey@ul.ie

ABSTRACT

Lack of flexibility and the high cost of Commercial-Off-The-Shelf (COTS) Discrete Event Simulation (DES) software, are two of the main factors that deter organizations from using DES in their decision support. Open Source (OS) simulation tools can provide an adequate solution, since the user has full access to the source code and thus gets the chance to make modifications at multiple levels. ManPy (Manufacturing in Python) is a new OS library of DES objects implemented in SimPy. ManPy's scope is to provide modellers with generic, highly customizable OS simulation objects that can be connected to form a model in the same fashion of COTS simulation packages. Extensibility is one of the requirements of ManPy. In this paper we present an extension of ManPy objects, to allow the modelling of production lines, where the products flow in batches, can be decomposed into sub-batches and reassembled to the original batch.

Keywords: Discrete Event Simulation, Open Source, SimPy, ManPy

1 INTRODUCTION

Discrete Event Simulation (DES) is one of the most widely accepted and used operational research (OR) methodologies (Shannon, 1998; Karabuk and Grant, 2007; Tako and Robinson, 2010). Although DES can be computationally expensive (Fujimoto, 1990; Can and Heavey, 2012), it is often preferred to mathematical methods due to its flexibility (Seila, 1995; Banks, 2000; Tako and Robinson 2012).

Commercial-Off-The-Shelf (COTS) simulation software packages have contributed in making DES widely applicable since they significantly reduce the programming effort. Nevertheless, the user needs still to be a DES expert in order to model all the peculiarities of an individual system (Kuljis and Paul, 2001). Moreover, COTS DES software can be difficult to use due to lack of flexibility in large and complex applications (Pidd, 2004; King and Harrison, 2010). The high cost of COTS simulation software can also be an issue, especially for Small and Medium Enterprises (SMEs). This cost does not consist only of, the considerably high, licensing fee, but also of other factors such as training, the addition of plug-ins (i.e. optimization) and maintenance upgrades (McLean and Leong, 2001). Reusability of DES models and components (Paul and Taylor, 2002; Balci et al. 2011) is also less applicable when COTS software is used, since non-renewal of a license can mean that the COTS models already developed cannot be used in the future.

Open-Source (OS) DES software could potentially provide solutions to some of the above problems. The lower cost is one obvious benefit since OS permits the use of the software without a licensing fee (Fogel, 2005; Di Penta et al. 2010). Reusability can also be enhanced since past DES objects and models will always be available for future reuse. In terms of speed, due to the absence of licensing issues, OS offers the capability of free distribution to allow web-based simulation (Byrne et al 2010).

However, most OS DES projects seem to have failed in gaining attention and remaining active (Dagkakis et al 2013a). From our review, it was noted that one issue is that most of this software requires that the end user is an experienced programmer in order to model even simple cases. In this paper we present the extension of a new OS simulation project called ManPy ("Manufacturing in Python") enriching it with new objects in order to be able to model production lines, where the products flow in batches, can be decomposed into sub-batches and reassembled to the original batch. The scope of this work was to establish the extensibility of our simulation engine, which is part of a broader OS decision support platform called "simulation based application Decision support in Real-time for Efficient Agile Manufacturing" (DREAM). The DREAM project (http://dream-simulation.eu/ accessed 11 November 2013) started in October of 2012 and its objective is to address the multi-faceted problems that deter manufacturing enterprises from more widely adopting simulation based decision support applications. ManPy is the simulation engine developed within the DREAM platform, designed to co-operate with other DREAM modules and provide tailored solutions for organisations. Nevertheless, the DREAM architecture is modular and in essence ManPy is also a standalone OS DES project result.

The outline of the remaining of this paper is as follows: In the following section we present ManPy, its scope, architecture and current progress. In section 3, through an example, we illustrate how the ManPy tool can be extended. In section 4 we compare the ManPy extension against a COTS simulation package and measurable and non-measurable results and insights gained are described. We end the paper with conclusions and suggestions for further research.

2 PRESENTATION OF MANPY

ManPy stands for "Manufacturing in Python" and is a layer of generic, highly customizable DES objects implemented in SimPy (http://simpy.readthedocs.org/en/latest/ accessed 11 November 2013). ManPy, an ongoing project within the FP7 DREAM project is currently mature enough to be used and extended by third parties. It was launched in GitHub as an OS project in November 2013 (https://github.com/nexedi/dream accessed 11 November 2013). This OS repository contains also documentation to facilitate contribution to the project.

2.1 Review of OS DES

From our review of OS DES tools (Dagkakis et al 2013a) we found that most of these projects fail to remain active and attract contributors to these projects. Two exceptions are the OS DES tool OMNeT++ (http://www.omnetpp.org/ accessed 11 November 2013) that is primarily targeted at network simulations and has an active OS community (Varga and Hornig, 2008) and JaamSim (https://github.com/AusencoSimulation/JaamSim accessed 11 November 2013) that provides a very impressive 3d Graphical User Interface (GUI) (King and Harrison, 2010). While OMNeT++ is primarily aimed at network modeling, it has also been used for manufacturing (Bause et al. 2010). Our belief is that OMNeT++ is popular because it is targeted to the domain of network modelling, which would attract programmers, who would also be users of this simulation software. JaamSim has only been recently released and it is too early to state if it will attract the same amount of developers.

Among the 23 OS DES projects we reviewed was also SimPy, which we decided to develop this current work on. SimPy stands for "Simulation in Python" and according to its authors it is "a process-based discrete-event simulation framework based on standard Python". Python provides merits for efficient and clean programming with great capabilities in list processing and a very flexible type casting, which make it ideal for DES. That does not imply that other computer languages do not provide efficient type casting, but the easiness that Python offers in this aspect, makes it more attractive even to users that are not computer experts. The trade-off is that Python, as a scripting language, is by default slower than static languages such as C++ or Java (Dawson, 2002), even though there have been attempts to make SimPy faster (Bahouth et al 2007). SimPy is quite active as a project and it is licensed under LGPL, which makes it feasible to be also used in proprietary projects (McGowan, 2005). The requirements of DREAM dictate that the platform will be OS, but it should also provide the possibility of being expanded to a tailored and closed solution for a company, so a license like LGPL or a more permissive license is essential for our needs. The fact that other interesting tools, like JaamSim, are published under GPL, would hinder their use as the basis of DREAM. The criteria listed above are quantified in Dagkakis et al. (2013a).

From our use of SimPy we established that a user needs to be a proficient programmer even to build simple models. We wanted to provide something new, developing manufacturing objects that would be well defined and a user can connect like "black boxes", in the same fashion that COTS DES packages work. Moreover, we wanted to give the chance to more advanced users to customize the objects or even create completely new ones and incorporate them into the repository. ManPy as a simulation engine is exclusively written in Python and takes advantage of SimPy's efficient use of Python generators that is achieved via the SimPy.Process class. ManPy uses the yield statements that SimPy offers, which give a means of managing the sequence that take control of a program during a simulation run. The scope is not to imitate or replace SimPy, but to provide something different and new in the OS DES field to support manufacturing first, but also logistics and services in the future. ManPy currently uses SimPy2 (http://simpy.sourceforge.net/old/ accessed 11 November 2013) but it is in our plans to use the recently released SimPy3.

2.2 ManPy User Levels

From our interaction with DREAM industrial partners we identified three possible categories of ManPy users:

- **Super User**: she/he can get deep into ManPy code, customize objects or create completely new ones and add them to the repository. She/he needs to have good coding and modeling skills.
- **Industrial Engineer**: she/he can use tailored ManPy objects to connect them and create a model. Limited coding experience is required at this level, but good modeling skills are essential.
- **End User**: she/he takes a ManPy model which is tailored for her/his needs and needs just to specify certain values using forms, drop-down menus etc. No software or modeling experience required.

2.3 ManPy Architecture

DES is generally a very appropriate paradigm where object oriented programming can be used (Smith, 2011). It is indicative that the ALGOL based simulation language Simula was historically the first that introduced the class concept (Rentsch, 1982). The current architecture of ManPy is shown in Figure 1.

On the top of Figure 1 (and lower level of architecture to be precise) lies SimPy, which ManPy adopted as its basis. At the next level we have five categories of abstract classes which are important in ManPy. These classes offer generic methods that all ManPy objects inherit and potentially override. These are:

- **CoreObject**: all the stations which are permanent for the model. These can be servers or buffers of any type and also entry and exit points.
- **ObjectInterruption**: all the objects that can affect the availability of another object. For example failures, scheduled breaks, shifts etc.
- **Entity**: all objects that get processed by or wait in CoreObjects and they are not permanent in a model. For example parts in a production line, customers in a shop, calls in a call centre etc.
- **ObjectResource**: all the resources that might be necessary for certain operations of a CoreObject. For example repairman, operator, electric power etc. An ObjectResource is

necessary in modeling when two or more CoreObjects compete for the same resource (e.g. two machines competing for the same operator).

- **Auxiliary**: These are auxiliary scripts that are needed for different simulation functionalities. Unlike the other categories described here, auxiliary classes do not inherit from one parent class, even though it is depicted in such a way in Figure 1 for reasons of coherence.

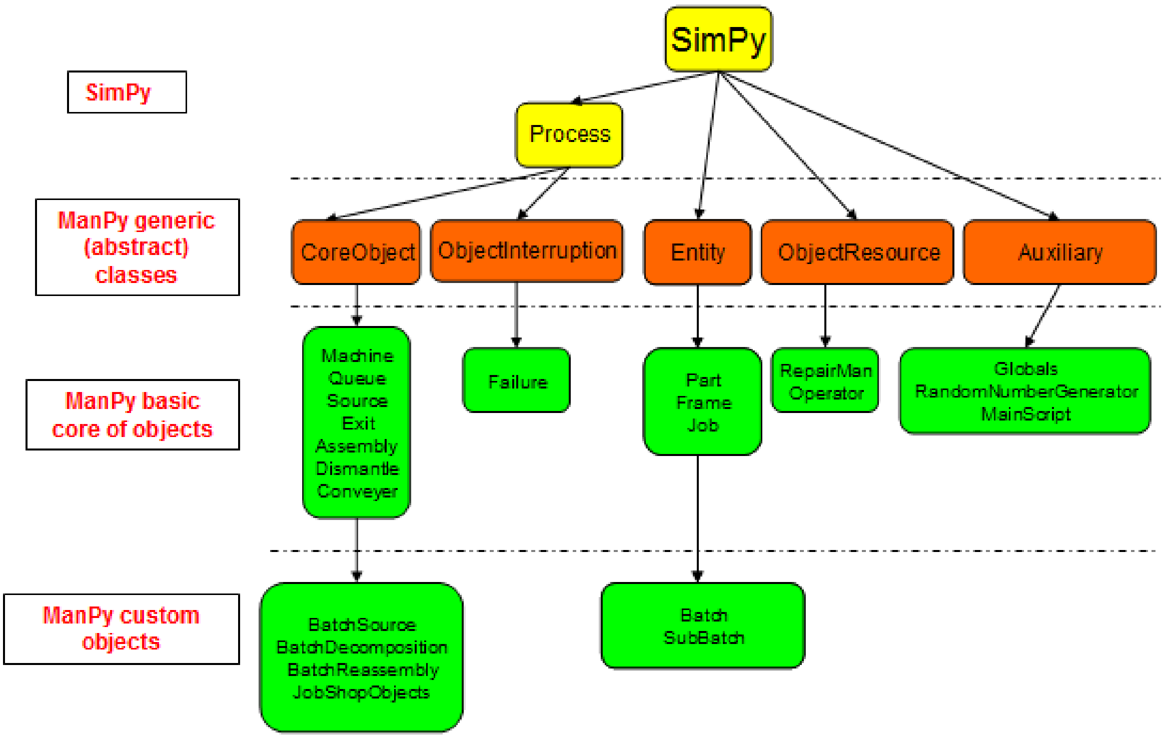


Figure 1 *ManPy Architecture*

In Figure 1 we can see that only the CoreObject and ObjectInterruption categories are the ones to inherit from the SimPy.process. It is in the ManPy philosophy, in order to achieve a cleaner process orientation that the CoreObjects (i.e. the stations) are the ones to handle how Entities and ObjectResources move in simulation time. Also note that the naming conventions of the generic classes are under review and we are in the process of upgrading these names into something more generic.

The other two levels show current concrete classes of ManPy that may be either generic or customized. The DREAM pilot cases are used to dictate what new objects are needed in the simulation platform. In this article we will demonstrate how the core was extended with new objects in order to be able to model production lines, where the products flow in batches, can be decomposed into sub-batches and reassembled to the original batch. We aspire in this project that both the core and the custom objects layers will be actively expanded with new DES objects.

The architecture described above, aims to be short and complete. Short because having a long list of generic concepts, principles and methods makes an OS project less attractive for contributors since the initial learning curve is longer. Complete means that every object needed to model complex operations can semantically fall into one of the generic classes. The use of ManPy in DREAM pilot cases, which cover a quite indicative range of operational systems, will be used to prove this concept. Following this approach the systems are reduced to stationary objects (CoreObjects) that exchange mobile objects (Entities). Third party resources may be needed in these transactions (ObjectResources), while the availability of objects at any point of the simulation time may be affected by certain processes (ObjectResources). Auxiliary classes and methods help in the binding of all the ManPy objects into a functional simulation model.

In order to give the functionality to the objects to communicate like “black boxes”, every object category has to carry attributes and implement methods that follow a certain ManPy naming convention. In this way for example a Machine object can take an Entity from another CoreObject no matter if this is a Queue, an Assembly or another Machine. Python's flexibility in type casting has proven very helpful in this approach. More information about the categories of methods can be found in Dagkakis et al (2013b) and an even more detailed description is available in the ManPy documentation in GitHub. Here we will mention some methods that were used in the new objects implemented for the specific work that is presented in this article. These are all methods of the CoreObjects:

- ***__init__***: this is the python constructor method. This method is run when the instance is created.
- ***initialize***: this method initializes the object for a simulation replication. It should not be confused with the constructor above. The constructor is run only in the creation of the object, while initialize must be invoked in the beginning of every replication.
- ***canAccept***: returns true if the object is in a state to receive an Entity. The logic depends on the type of the object. For example in a Queue the capacity might need to be checked, while an Exit object might always be in the state of receiving an Entity.
- ***createEntity***: creates new Entities in the CoreObject. Most usually used by entry points such as Source, but every object can potentially use it.
- ***haveToDispose***: returns true if the object is in a state to give an Entity. The logic depends on the type of object. For example a Queue may need to check only if it does hold one or more Entities, while a Machine might need to check if the Entity that it holds has ended its processing.
- ***canAcceptAndIsRequested***: returns true only when both conditions are satisfied: the object is in the state of accepting an Entity and also another object is requesting to give one Entity to it. Only when this method returns true the main simulation logic of the object is started.
- ***run***: this is a generator method and it is the one where the logic of the progress of the object in simulation time is implemented. For this reason run requires that the user has also knowledge of SimPy in order to customize its behaviour.

In the next section we will see how such methods were overridden for the implementation of new objects that carry customized simulation logic.

3 EXTENSION OF MANPY

Clean extensibility is one of the key success factors of ManPy in order to attract users and contributors and create an OS DES community. Our approach to establish this concept was to have a new super user who would be different from the original author of ManPy. This user would try to learn the ManPy platform, implement, test and document the new objects. We cannot claim that this is exactly a third party, since this user would have the chance to have face to face communication with the ManPy author in order to ask questions, but we regard it as a first step for the validation of the ManPy extensibility concept. Further experiments of this nature that would include more users and models are left for future work.

The original set of ManPy objects could not handle cases of production lines, where the products flow in batches and can be decomposed into sub-batches and reassembled to form the original batch. After the user read through the ManPy documentation and became accustomed to using the code there was a meeting with the ManPy Author in order to define the new objects needed, their attributes and methods. These were:

- **Batch**: Entity that holds a number of units and can be broken into a number of other Entities (sub-batches).
- **SubBatch**: Entity that contains a number of units and is derived from a parent Batch. In its attributes it also holds the parent Batch.
- **BatchSource**: CoreObject that inherits from Source and creates Entities that hold a number of units. It overrides the *__init__* method in order to accept the number of units as arguments and also the *createEntity* method so that it creates the Entity giving this number of units.
- **BatchDecomposition** – CoreObject that takes a batch and decomposes to sub-batches. It overrides the CoreObject’s *__init__*, *initialize*, *canAccept*, *haveToDispose*,

canAcceptAndIsRequested and run methods in order to implement the new decomposition logic.

- **BatchReassembly** – CoreObject that takes a number of sub-batches and reassembles the original batch. It overrides the CoreObject's *__init__*, *initialize*, *canAccept*, *haveToDispose*, *canAcceptAndIsRequested* and run methods in order to implement the new logic.

The two new Entity types required just the addition of new attributes and BatchSource was a customization of an already existing ManPy object. These have been quite simple to implement and the whole development of them took a couple of hours. As expected, the two new CoreObject types were more difficult to code and debug and the process consumed one working day for each.

It is very important to state that in all of this coding process there was no need to make modifications to the code in the already existing objects. The underlying code was considered as a "black-box" and the new object could be incorporated into the repository and be used in models that contain also objects from this "black-box".

4 VERIFICATION, COMPARISON AND DOCUMENTATION

4.1 Verification

DES code is notoriously difficult to debug and verify (Sargent, 2001; Balci, 2003). The fact that there are processes running in parallel and simultaneously taking control of the program makes it cumbersome for the programmer to keep a track on model behaviour. ManPy is not a complete project yet, but it does offer some assets for verification. First of all, the objects implement an *outputTrace* method which is used in order to output the trace of the run into an Excel spreadsheet. All ManPy objects output to the same Excel file and the events are sorted in increasing timestamp. The trace is essential for debugging. To run a model that is believed to be verified, it should be turned off since it slows the program significantly.

Other tools that are used during implementation are version control (Fischer et al 2003) through Git (http://git-scm.com/ accessed 11 November 2013) and unit-testing (Zhu et al 1997). The former enhances the co-operation of several developers in the same project and unit-testing is used to make sure that a developer does not damage work previously verified. Bug tracking (Just et al 2008) has not been used in ManPy related code before it went public, since a limited number of developers were involved and issues could be reported via e-mail, but it is something to be considered when the project gets external users. ManPy offers also a web-based GUI that is being implemented in parallel for the needs of DREAM platform, but does not provide the feature of real time animation which could also significantly enhance the verification process (King and Harrison, 2010). This is left for future work.

In addition to the above, it is our standard practice to use a COTS DES package in order to verify new ManPy objects. Such COTS software is around for many years and has already been used by many users worldwide so they have achieved a higher level of code accuracy than the more recently developed ManPy. Building simple models with the new objects in both ManPy and a COTS DES package and achieving the exact same results in deterministic situations, is our way of gaining confidence in our new objects. Plant Simulation by Siemens (http://www.plm.automation.siemens.com accessed 11 November 2013) was selected, as a package that we have extensively used in the past (Dagkakis et al 2013b).

After the new ManPy objects were implemented two simple models were made both in ManPy and Plant. Figures 2 and 3 show the implemented Plant models. The first one has a BatchSource (BS) that creates batches assigning them a batch id and a number of units. Then the batches are buffered (BDQ) before they go into a BatchDecomposition object (BD) where they are broken into sub-batches and then they are sent into a station (S1) that processes them before they reach the Exit. In Figure 3 the models are identical until station S1 but then there is a buffer (Q1) followed by another station (S2) and a BatchReassembly object (BR) where the batch is reassembled before it is sent to a last station (S3) and finally reaches the Exit. In Plant new objects were required. Batch and SubBatch are new objects of Plant's MUs, BatchDecomposition inherits from Plant's PlaceBuffer and BatchReassembly from SingleProc. Additionally, in Plant the preceding buffer of the BatchDecomposition had also to be developed (BatchDecompositionQueue). All of these objects had to customize their Entrance, Exit and Observer methods. More information about prefabricated Plant types, such as MUs, SingleProc and PlaceBuffer that were extended in order to model the new objects, can be found in Bangsow (2010).

Using DREAM GUI (http://212.85.154.245:5000/static/index.html accessed 22 January 2014) the user can drag and drop the objects in the interface, connect them with arrows, set attributes, run the model and get the results with no need to code at all. DREAM GUI is being developed in parallel and it is already linked to the original set of ManPy objects.

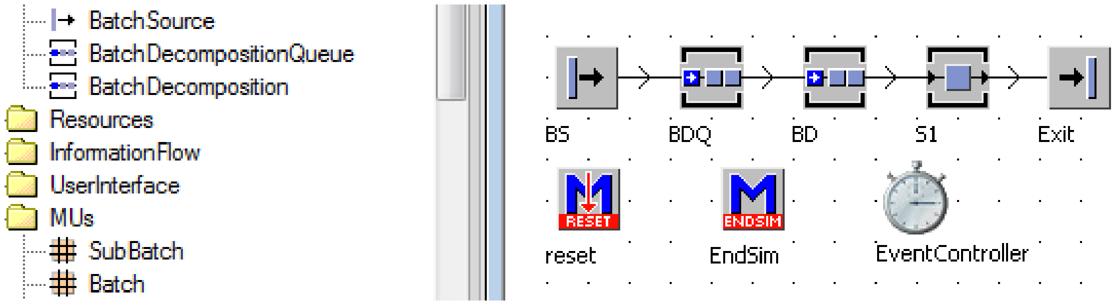


Figure 2 *Plant model to test the decomposition of the Batches*

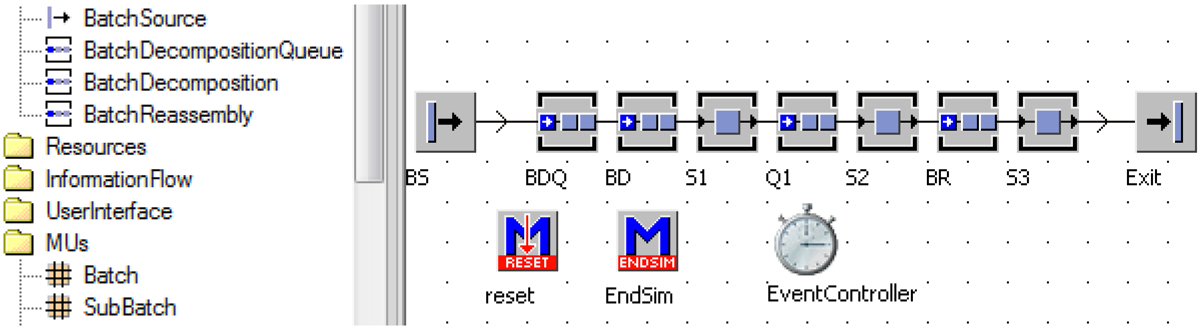


Figure 3 *Plant model to test the decomposition and reassembly of the Batches*

Since the exercise presented here involved a Super User, it was considered desirable that this user builds the models programmatically. To programmatically build a model of existing and extended objects in ManPy a main script has to be coded. This is the one that ties all the objects in a model and runs the simulation. A ManPy main script defines, creates and connects the objects, runs the simulation replications and outputs results. In this manner ManPy remains as a standalone simulation engine, facilitating embedding the simulation engine into decision support systems.

4.2 Comparison

The models were run using sets of different attributes (number of units in batches, number of sub-batches that a batch is broken into, capacity of the queues, processing times) and after debugging we established that the results were identical up to the last decimal digits in the deterministic case. This comparison leads to the verification that the logic of the objects is implemented correctly.

In terms of speed ManPy is slower than Plant. This issue was demonstrated in Dagkakis et al (2013b) and, regarding the computational efficiency, experiments with the new objects gave results that are similar to the ones presented in that article. Note that all the experiments were run in the same machine. Nevertheless, ManPy can be less expensively used in a cluster of PCs, since there is no license required. This can enhance the speed significantly using a web-based distributed simulation in cases where we have multiple replications of stochastic models or scenario analysis (Byrne et al 2010). It is in the scope of DREAM to research such methodologies in order to provide a high performance DES framework.

About modelling effort we do not have measurable results, but we identified that we used different ways to model the same logic in Plant and ManPy. For example, the Plant modelling approach required that the preceding buffer of Batch decomposition is also customized. Plant is an extremely efficient package that gives great flexibility to a user that has some coding ability. However, we found that the fact that in ManPy all code is available would many times meliorate both modelling and debugging.

5 CONCLUSIONS AND FUTURE WORK

In this article we demonstrated how an OS DES project called ManPy was expanded in order to be able to cover custom cases. The procedure we followed is regarded as the first step towards the validation of the ManPy extensibility concept, which is crucial for the success of our project. A new super user got trained to the platform and was able to contribute adequately fast. On the other hand, he had the chance of having face to face communication with the ManPy author, which reduced the time required for him. It is of future work to try to establish the extensibility and reusability concepts with remote users. If it is possible to create an active OS DES community, where programmers, modellers and researchers can efficiently exchange, document and understand DES objects, then OS DES will become a strong competitor of COTS DES packages.

ACKNOWLEDGMENTS

The research leading to the results presented in this paper has received funding from the European Union Seventh Framework Programme (FP7-2012-NMP-ICT-FoF) under grant agreement n° 314364.

REFERENCES

Bahouth A, Crites S, Matloff N, Williamson T (2007) Revisiting the Issue of Performance Enhancement of Discrete Event Simulation Software. *Proceedings of the 40th Annual Simulation Symposium*, pp 114–122. doi: 10.1109/ANSS.2007.36.

Balci O (2003) Verification, validation, and certification of modeling and simulation applications: verification, validation, and certification of modeling and simulation applications. *Proceedings of the 35th Conference on Winter Simulation: Driving Innovation*, pp 150–158.

Balci O, Arthur JD, Ormsby WF (2011) Achieving reusability and composability with a simulation conceptual model. *Journal of Simulation* **5(3)**: 157–165.

Bangsow S (2010) *Manufacturing simulation with plant simulation and simtalk: usage and programming with examples and solutions.* Springer.

Banks J (2000) Introduction to simulation. *Proceedings of the Winter Simulation Conference 2000*. Ieee, pp 9–16.

Bause F, Buchholz P, Kriege J, Vastag S (2010) A Simulation Environment for Hierarchical Process Chains Based on OMNeT++. *Proceedings of the Winter Simulation Conference 2011*. Ieee, pp 9–16.

Byrne J, Heavey C, Byrne PJ (2010) A review of Web-based simulation and supporting tools. *Journal of Simulation Modelling Practice and Theory* **18(3)**: 253–276. doi: 10.1016/j.simpat.2009.09.013.

Can B, Heavey C (2012) A comparison of genetic programming and artificial neural networks in metamodeling of discrete-event simulation models. *Journal of Computers & Operations Research* **39 (2)**: 424–436.

Dagkakis G, Heavey C, Papadopoulos CT (2013a) A Review of Open Source Discrete Event Simulation Software. *Proceedings of the 9th Conference on Stochastic Models of Manufacturing and Service Operations*, pp 27 – 35.

Dagkakis G, Heavey C, Robin S, Perrin J (2013b) ManPy: An Open-Source Layer of DES Manufacturing Objects Implemented in SimPy. *Proceedings of the 8th EUROSIM Congress on Modelling and Simulation, EUROSIM2013.*

Dawson B (2002) Game scripting in Python. *Proceedings of GDC.*

Fischer M, Pinzger M, Gall H (2003) Populating a Release History Database from version control and bug tracking systems. *Proceedings of the International Conference on Software Maintenance (ISCM) 2003*, pp 23–32. doi: 10.1109/ICSM.2003.1235403.

Fogel K (2005) *Producing open source software: How to run a successful free software project.* O'Reilly Media, Inc.

Fujimoto R (1990) Parallel discrete event simulation. *Journal of Communication ACM* **33(10)**: 30 – 53.

Just S, Premraj R, Zimmermann T (2008) Towards the next generation of bug tracking systems. *IEEE Symposium on Visual Languages and Human-Centric Computing 2008*, pp 82–85. doi: 10.1109/VLHCC.2008.4639063.

Karabuk S, Grant F (2007) A common medium for programming operations-research models. *Journal of Software, IEEE* **24(5)**: 39–47.

King D, Harrison H (2010) Discrete-event simulation in Java: a practitioner's experience. *Proceedings of the 2010 Conference on Grand Challenges in Modeling & Simulation l*, pp 436–441.

Kuljis J, Paul RJ (2001) An appraisal of web-based simulation: whither we wander? *Journal of Simulation Practice and Theory* **9(1-2)**: 37–54. doi: 10.1016/S0928-4869(01)00032-5.

McGowan D (2005) Legal aspects of free and open source software. *Journal of Perspectives on Free and Open Source Software,* pp 1–29.

McLean C, Leong S (2001) The expanding role of simulation in future manufacturing. *Proceedings of the 33nd Winter Simulation Confenrence,* pp 1478–1486.

Paul RJ, Taylor SJE (2002) What use is model reuse: is there a crook at the end of the rainbow? *Proceedings of the Winter Simulation Confenrence 2002,* pp 648–652. doi: 10.1109/WSC.2002.1172943.

Di Penta M, German DM, Guéhéneuc Y-G, Antoniol G (2010) An exploratory study of the evolution of software licensing. *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering,* pp 145–154.

Pidd M (2004) Simulation worldviews: so what? *Proceedings of the 36th Winter Simulation Conference,* pp 288–292.

Rentsch T (1982) Object oriented programming. *Journal SIGPLAN Not.* **17(9)**: 51–57.

Sargent RG (2001) Some approaches and paradigms for verifying and validating simulation models. *Proceedings of the 2001 Winter Simulation Conf*erence, pp 106–114. doi: 10.1109/WSC.2001.977251.

Seila A (1995) Introduction to simulation. *Proceedings of the 1995 Winter Simulation Conference*, pp 7–15.

Shannon RE (1998) Introduction to the art and science of simulation. *Proceedings of the 1998 Winter Simulation Conference*, pp 7–14. doi: 10.1109/WSC.1998.744892.

Smith B (2011) *AdvancED ActionScript 3.0: Design Patterns,* Springer.

Tako A a., Robinson S (2010) Model development in discrete-event simulation and system dynamics: An empirical study of expert modellers. *European Journal of Operational Research* **207(2):** 784–794.

Tako A a., Robinson S (2012) The application of discrete event simulation and system dynamics in the logistics and supply chain context. *Journal of Decision Supportort Systems* **52(4)**: 802–815.

Varga A, Hornig R (2008) An Overview of the OMNeT++ Simulation Environment. *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, pp 60:1–60:10.

Zhu H, Hall P a. V., May JHR (1997) Software unit test coverage and adequacy. *Journal of ACM Comput Surv* **29(4):** 366–427. doi: 10.1145/267580.267590.

AUTHOR BIOGRAPHIES

GEORGIOS DAGKAKIS received his M. Eng in Electrical Engineering and Computer Science from the Aristotle University of Thessaloniki in 2007. He completed his M. Sc. in Informatics and Management at the same University in 2009. He is currently a PhD student in the University of Limerick. His expertise is in Discrete Event Simulation modelling.

IOANNIS PAPAGIANNOPOULOS is an Electrical and Computer Engineer. After graduating from Aristotle University of Thessaloniki he undertook doctoral studies in the field of Heat Transfer in Electronics with a broader interest in simulation techniques and programming. He recently commenced postgraduate studies in the field of Discrete Event Simulation.

CATHAL HEAVEY is an Associate Professor in the Department of Design & Manufacturing Technology at the University of Limerick. He is an Industrial Engineering graduate of the National University of Ireland (University College Galway) and holds a M. Eng.Sc. and Ph.D. from the same University. He has published in the areas of queuing and simulation modelling. His research interests includes, simulation modelling of discrete-event systems; modelling and analysis of supply chains and manufacturing systems; process modelling; component-based simulation and decision support systems

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

MODELLING THE DENTAL WORKFORCE IN SRI LANKA

Professor Sally Brailsford

School of Management
University of Southampton
Southampton SO17 1BJ, U.K.
s.c.brailsford@soton.ac.uk

Dr Dileep De Silva

Dental Services Human Resources
Ministry of Health
Colombo, Sri Lanka
dileepdenta@yahoo.com

ABSTRACT

This paper describes a system dynamics model of the dental workforce in Sri Lanka. The model represents the career progression of dentists, from recruitment and training at the university dental school, through different types of employment through to retirement. The aim of this model, after it was augmented with a separate model depicting future demand scenarios for dental care (not presented here), was to allow the Sri Lankan government to plan better provision of state-funded dental care and the future university intake of dental students.

Keywords: System Dynamics, workforce planning, dental health

1 INTRODUCTION

Sri Lanka is an island nation in the Indian Ocean with a population of 20 million. Although it is classed by the World Bank as a developing country, its social indicators are impressive for a low-income country, as evidenced by its infant mortality rate of 9.7 per 1,000 live births, its life expectancy at birth of 75.73 years and adult literacy rate of 90% (Central Bank of Sri Lanka, 2009). In many respects, the healthcare system in Sri Lanka is not unlike the UK's National Health Service. Basic medical and dental services are provided free at the point of delivery, and the system is funded by taxation. As in the UK, dentists in Sri Lanka can work in either the private or state sectors, and many do both. State employment is, in theory at least, guaranteed to all Sri Lankan-trained dental surgeons on completion of their training.

However, one respect in which Sri Lanka differs from the UK is that all education, including university education, is provided completely free of charge to the student. No doubt this contributes significantly to the high literacy rate. This includes medicine and dentistry, among the most expensive university courses in any country, although a cap is placed on the number of places the Sri Lankan Government funds. For various political reasons, from the early 1990's onwards the Sri Lankan Government funded too many university places in dentistry and as a result was then unable to employ all its qualified dentists. By the start of 2010 there were over 250 dental surgeons awaiting state employment, nearly 25% of the total number available. However there were no jobs for these people. Many either left to practice overseas, or tried to set up in private practice, but the latter is of course difficult for a newly qualified dentist with no established reputation to attract patients or the money to buy a practice. Moreover, very few dentists wished to work in the more remote parts of the country, preferring to remain in the capital city Colombo where people are wealthier and the opportunities for private practice are greater. In 2010 some rural districts had less than three dental clinics per 100,000 population, compared with ten in Colombo, 46 in the UK and 58 in the USA.

This under/unemployment among dental surgeons was increasing, due to the political agenda and a lack of coordination between the Ministry of Higher Education (who pay for the training) and the Ministry of Health (who employ the dentists). At the same time the University Grant Commission, which decides the number of undergraduates to be trained, was under continuous political pressure to increase the intake of dental students. This was therefore a highly contentious political issue about the use of taxpayers' money. This would be contentious enough in the UK, but in Sri Lanka the problem was aggravated by the desperate need for improved population access to dental care. Less than 10% of the Sri Lankan population visited a state-funded dentist in 2007; 40% of 12-year-olds had decayed teeth; 90% of the adult population had gum disease, and 51% of the 1.8 million dental visits per year were for dental extractions. The dentist to population ratio recommended by the World Health Organization is 1:4500, but the Sri Lanka ratio in 2010 was 1:13,500 (De Silva, 2012).

2 A DENTIST TURNS MODELLER

Dileep De Silva, co-author of this paper, was himself one of these "underemployed" dentists who was unable to find a state-funded clinical post. However, unusually, he had an MBA, and this enabled him to obtain administrative employment as a civil servant within the Ministry of Health. He soon realized that there was a major problem with the Sri Lankan system and that this problem was gradually getting worse. Like many of his colleagues he then chose to go overseas rather than remain at home, but rather than practice as a dentist, he decided to further his education and undertake a PhD. While searching on the internet for PhD programmes which might enable him to tackle this problem, he came across the healthcare modelling work at the University of Southampton. At that point he had no previous knowledge or experience of management science, let alone simulation modelling. However he was a fast learner! – and the rest, as they say, is history…

2.1 The dental career structure in Sri Lanka

Sri Lanka has only one Dental School, at the University of Peradeniya in Kandy, the second largest city in the country. Only a very small number of foreign-qualified dentists (a total of 36 in 2010) work in Sri Lanka. In 2010, the qualified dental workforce was around 1,500, and the annual intake of dentistry students was 80. The normal duration of a dentistry degree is four years, although in practice a small number of students take longer than this, or drop out for various reasons. After graduating, in the past most newly-qualified dental surgeons would immediately obtain state (i.e. "NHS") employment with the Ministry of Health, usually working in a hospital. Other career options, chosen by a small number of dentists, include joining the defence sector as a military dental surgeon, or (for a very few) remaining in academia. Traditionally, a typical career would involve working initially for a few years solely in the state sector, and then, after gaining experience and establishing a local reputation, setting up in part-time private practice. All state-employed dental surgeons, including those employed in the defence sector or by the university, are permitted to work in part-time private practice after normal hospital working hours. The majority do this: the opportunity to work in private practice is the only way for most people to augment their salary above the basic level paid by the state. However, as described above, in recent years increasingly many newly qualified graduates had been forced to go overseas to gain work experience.

Sri Lankan dentists can (and do) switch between these different career options over the course of their working lives, subject to employment contract restrictions in the chosen career pathway. For example, dentists in the defence sector are bonded to serve for a certain minimum time (normally 12 years), and thus are not entitled to change job category during the bonded period. Some people may leave the profession altogether. Others dentists may choose further specialist training in one of five areas: maxillo-facial surgery, orthodontics, restorative dentistry, oral pathology and community dentistry. There is no mandatory retirement age in the private sector, although in the state sector it is 60 years; but generally people quit state or defence employment and just retain their part-time private practice, gradually reducing their hours, once they reach the age of about 50-55.

2.2 Choice of modelling methodology

System dynamics was chosen as the modelling approach because it lends itself very well to modelling the progression of dentists through the various career stages. As described briefly in section 2.3 below, many similar system dynamics models exist in the literature for workforce planning.

System dynamics (SD) is a simulation modelling methodology which can incorporate both qualitative and quantitative approaches. A quantitative SD model consists of a set of stocks or "levels" connected by pipes through which material flows in a continuous manner. The rate of flow is governed by valves. A domestic water system is a reasonable analogy, where the tank, bathtub and washbasins represent stocks, the pipes between them represents the flows, and the rate of flow is governed by taps, as shown (highly simplistically) in Figure 1.

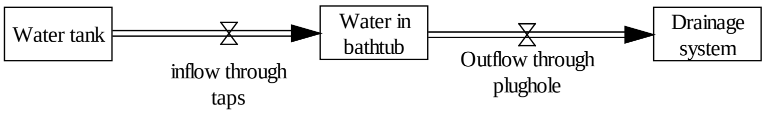


Figure 1 *A very simple stock flow model*

Mathematically, such models essentially consist of a set of ordinary differential equations representing the change in the stocks over time. In SD software, these ODEs are solved numerically by a discretization process with time-step *dt*. They therefore have the potential to be very complicated: the equations describing the rates of flow can include delays, influences from external parameters (known as auxiliary variables) which can take a wide variety of formats including graphical functions and empirical data, and many other refinements. For example, in the bathtub in Figure 1, the rate of inflow through the taps could be some function of the current water level and the prospective bather's personal opinion of a desirable depth. SD is regarded as a simulation approach, although SD models are not generally stochastic. This has the great benefit that multiple iterations are not required and so the models can be extremely quick to run. The "material" which flows through an SD model is continuous, and so the approach is not suitable for problems in which it is necessary to distinguish between individuals. It is however possible to use subscripted arrays, and these are often used in workforce models to represent age bands with differing attrition rates, for example.

The Sri Lankan model consists of stocks depicting the various career stages, the flows between them and the attrition from them as people leave the profession, die in service or retire. For example, the gradual retirement process described above can be modelled very appropriately by draining the stock of solely privately employed dentists at a given rate. Moreover, the transitions and flows between stocks are influenced by various factors such as the perceived job opportunities in particular sectors, again something which system dynamics can capture very nicely. The model was developed in *Stella*, also known as *iThink* (www.iseesystems.com/, accessed 17 November, 2013).

2.3 Brief overview of the literature

The aim of workforce planning, in any sector or industry, is to ensure that the right number of correctly qualified people are available at the right time at minimum cost to the organization. Many operational research techniques have been applied to model workforce planning systems. Mainly, methods such as optimization and simulation have been used in operational or tactical areas such as staff rostering, scheduling and capacity planning. However OR methods have also been used for more strategic workforce planning problems: the most common approaches are Markov chain models (usually implemented as a spreadsheet) and system dynamics. There is a considerable academic literature in this area, although we were unable to find any recent survey or review papers on OR for strategic workforce planning. In the mainstream OR literature, Gass (1991) provides a review of military applications, and Edwards (1983) gives a general review of OR in what was known in that pre-political correctness era as "manpower planning". However, a Web of Knowledge search on 17 November 2013 using the search string "*workforce planning model**" retrieved 1034 hits. A typical example is Grossler and Zock (2010) who use an SD ageing chain model to improve the recruiting and training process in a large German service provider in the field of logistics.

In the domain of healthcare, national bodies such as the UK's Centre for Workforce Intelligence (www.cfwi.org.uk, accessed November 17, 2013) who are charged with advising Government policy-makers about long-term training provision for health professionals, have developed a system dynamics model for future commissioning of university places for medical and dental training at the national level (www.cfwi.org.uk/publications/a-strategic-review-of-the-future-healthcare-workforce-informing-medical-and-dental-student-intakes-1/@@publication-detail, accessed November 17, 2013). More local-based workforce planning models have been developed by the Whole Systems Partnership (www.thewholesystem.co.uk/, accessed November 17, 2013), for example planning the medical and nursing workforce requirements for East Midlands NHS.

One of the best known dental workforce models is that of Bronkhorst (1995). This is an SD model produced for the Dutch Government to analyse supply and demand in the dental healthcare system in the Netherlands. The model contains about 440 state variables, and is one of the most complete and comprehensive scenario-based workforce models for dentistry. The model consists of five sub-models, each of which is a comprehensive model in its own right. On the demand side, the model includes demographic, pathological, sociological and economic processes. The supply side considers the availability of dental surgeons, dental hygienists and other factors influencing productivity. In this study, a wide range of future scenarios are explored and analysed, including different skill mix among dental staff, changes in the oral healthcare needs of the population, and disease patterns.

3 DESCRIPTION OF THE SRI LANKA MODEL

3.1 Qualitative modelling: influence diagrams

Although the model for Sri Lanka was a quantitative stock-flow model, representing the flow of dentists through the career stages described in section 2.1, the qualitative aspects of SD (influence diagrams) were also used to gain some understanding of what was going on in the system, and the forces that could lead to various outcomes and behaviours, both desirable and undesirable. In an influence diagram, pairwise relationships are depicted between key elements of the system, representing whether element A influences element B in the *same* direction (an increase in A leads to an increase in B, denoted by a plus sign) or in the *opposite* direction (an increase in A leads to a decrease in B, denoted by a minus sign), assuming everything else in the system remains unchanged. The aim is then to identify feedback loops – directed chains of causal links – which can either maintain system stability or lead to an unstable spiralling effect.

For example the influence diagram in Figure 2 depicts two competing effects. The inner loop has a stabilising effect (it is a balancing loop, in SD terms, as it contains an odd number of minus signs): increasing numbers of unemployed graduates and long waiting lists for state employment lead to a reduction in the training rate. However the outer loop has a spiralling effect: it is a vicious circle or reinforcing loop in SD terms, as it contains an even number (zero) of minus signs. Public knowledge that there are many unemployed dentists puts pressure on the Government to provide extra clinics and posts to staff them. However, if everything else in the system remains unchanged, the more new jobs there are, the more dentists will be needed and so the training rate must increase. This and other similar influence diagrams provide some insights into what is happening, but it is not possible to tell from this diagram alone which of these two effects will dominate in practice. For this, it is necessary to build a quantitative model in computer software, populate it with data, express the relationships between system elements in numerical terms, and then run the simulation model. Nevertheless, the stock-flow model is still able to capture the behavioural effects of employment prospects and government policy.

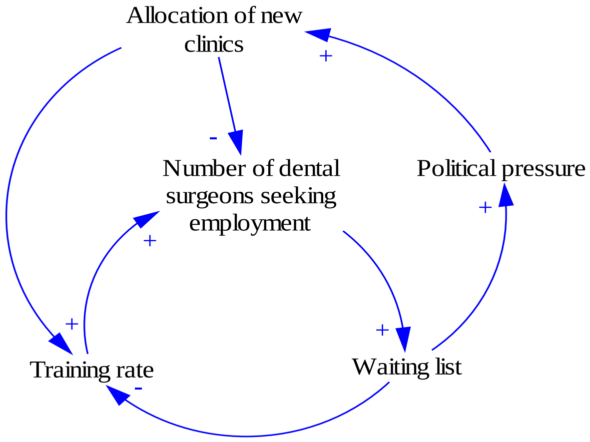


Figure 2 *Influence diagram demonstrating competing effects of long waits for employment*

3.2 The stock-flow model

A schematic of the full-time private practice part of the model is shown in Figure 3, and the complete career pathway is shown in Figure 4. A screen print of the entire Stella model is given in the Appendix.

After graduation from university, newly qualified dentists join a stock called "Entrants". At this point they transition to an employment category and work as dental surgeons (possibly in different capacities within any given category), until they exit the system due to retirement, attrition, or death. Furthermore, many people change their job category while remaining within the system. These people also join the Entrants stock, along with the very small number of foreign qualified dentists. In the past, as mentioned earlier, all graduates were guaranteed employment in one of the three Government-funded sectors: the university, the military, or the state sector, so the Entrants stock would have been empty most of the time and would have served solely as a device for routing flow around the system. However, due to the current lack of state employment, in 2010 this stock contained around 250 people who were either overseas or working in some job other than dentistry.

Since the private sector is more lucrative, most dentists augment their income by doing some part-time private practice. In former days when state employment was plentiful, many dental surgeons opted to become full-time private practitioners once they had saved up enough money to purchase a practice, and left their salaried job after a few years of employment. This option is not available to new graduates since they lack both the capital required to purchase an existing practice, and the experience and reputation to set up on their own. A few people attempt to set up a new ("*unestablished*") private practice, but this is risky and they may find it difficult to break even during the first few years. As a result they may become demotivated, as well as getting into financial difficulties, especially if they have taken out bank loans to start the practice. Also these new graduates are not long-term oriented, because their main aim is to gain the security of permanent salaried employment in the state sector. Thus only a few people enter full-time private practice as new graduates. In Figure 3, "Esta" (or "Est") denotes dentists with an *established* private practice, i.e. a strong reputation and a large number of patients, who are able to make a good living from private practice alone. Most of the stock "Esta FT PP" (*Established Full-Time Private Practitioners*) consists of dental surgeons who are nearing the end of their working lives. The stock "Unest FT PP" denotes the much smaller group of younger, recently-qualified dentists who are still seeking state employment.

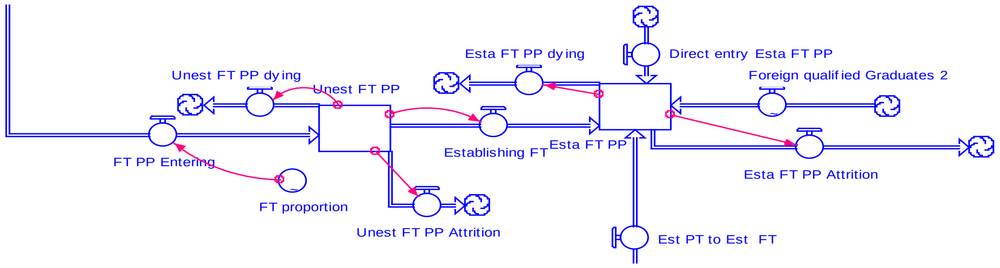


Figure 3 *The full-time private practice part of the model*

PP = private practice; PT = part-time; FT = full-time; Est(a) = established; Unest = unestablished

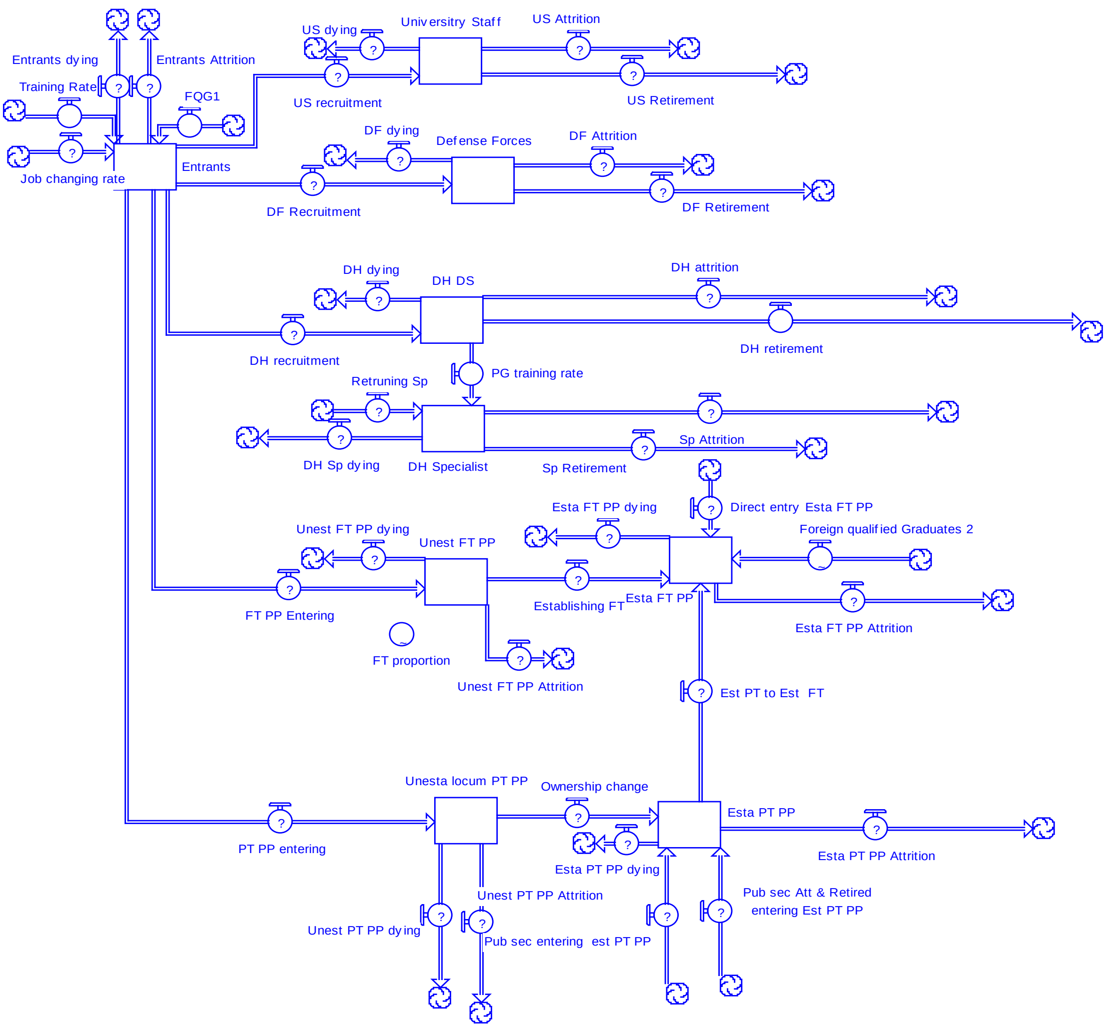


Figure 4 *Stella stock-flow diagram showing the entire system*

DH = Dept of Health (state sector);US = university sector; DF = defence forces; PP = private practice; Sp = specialist; PT = part-time; FT = full-time; Est(a) = established; Unest = unestablished (new)

3.3 Model outputs

The key model outputs are:

- the number of dental surgeons seeking state employment;
- the waiting time to secure state employment;
- the numbers of dental surgeons in different service categories;

- total dental surgeon hours available and total dental demand hours;
- percentage of demand met;
- dental surgeon to population ratios.

3.4 Data for the model

The model requires data on the initial numbers in each stock at time zero, plus the annual transition rates from stock to stock and a large number of auxiliary variables such as the ratio of state to private dentists, and the ratio of full-time to part-time employment (since "part-time" can mean anything from a couple of hours a week to nearly full-time). Populating a model such as this would be difficult enough in a developed country such as the UK, where there is a national register of qualified dentists and the demographic profile of currently practicing dentists is known. Even so, it would be a huge challenge to obtain data on the working hours of every dental practice, especially in a mixed economy such as the UK (and Sri Lanka) where dentists work both for the NHS and in private practice. In Sri Lanka, there was no national database of practicing dentists. Moreover, even in the UK it would be difficult to estimate the transition rates between full-time and part-time employment, NHS and private practice, attrition and retirement, and so on. The Centre for Workforce Intelligence model for medical and dental training intakes acknowledges this difficulty and the 2012 report concludes "*The CfWI provided advice on some significant gaps in dentistry data which led to the Health and Education National Strategic Exchange review group recommending improvements in the quality of data and a further review next year*". (www.cfwi.org.uk/publications/the-centre-for-workforce-intelligence-contributes-to-a-review-of-medical-and-dental-student-intakes, accessed 17 November, 2013).

De Silva therefore had to start from scratch. He had three advantages: firstly, virtually every dentist practicing in Sri Lanka would have graduated from the University of Peradeniya, and he had excellent data from the university on all its graduates since the very first cohort in 1946; secondly, he was technically employed by the Ministry of Health, so had access to data which would have been denied to most academic researchers; and thirdly, as a qualified dentist himself, he was able to gain the trust of his colleagues and explain the purpose of his study. Nevertheless he had some major disadvantages, of which by no means the least was that in 2008-09 a bitter civil war was still raging in the northern part of Sri Lanka and some areas were impossible to visit.

The data collection aspects of this research are recorded in detail in De Silva's thesis (De Silva, 2012). The first task included developing a national register of all practicing dentists! The entire data collection exercise was a major piece of investigative research which, in addition to archival research in the Ministry of Health, involved travelling the length and breadth of the country (apart from areas where it was not safe to travel due to the war), holding focus groups to get engagement and then sending out questionnaires. Current dental students were also surveyed. The response rate for the postal questionnaire, which was sent to all practicing dental surgeons in the country, was 72% (after two reminders): a remarkable achievement. Complete data on the model is available from the authors and is presented in De Silva (2012).

3.5 Model validation

The model was populated with historical data and run with the as-is values for all parameters from 1991-1995. It was then compared with existing data and found to be a pretty good fit, see Table 1. In addition, every stage of the model development was discussed with people who understand the system and so the model's face validity was considered to be extremely high.

Table 1 *Model validation results for 1991-1995*

Year	State employment seekers		Waiting time for state employment (months)	
	Model	***Actual***	***Model***	***Actual***
1991	29	26	7	6
1992	24	23	5	5
1993	37	35	8	7
1994	47	44	10	10
1995	56	55	12	12

4 USE OF THE MODEL

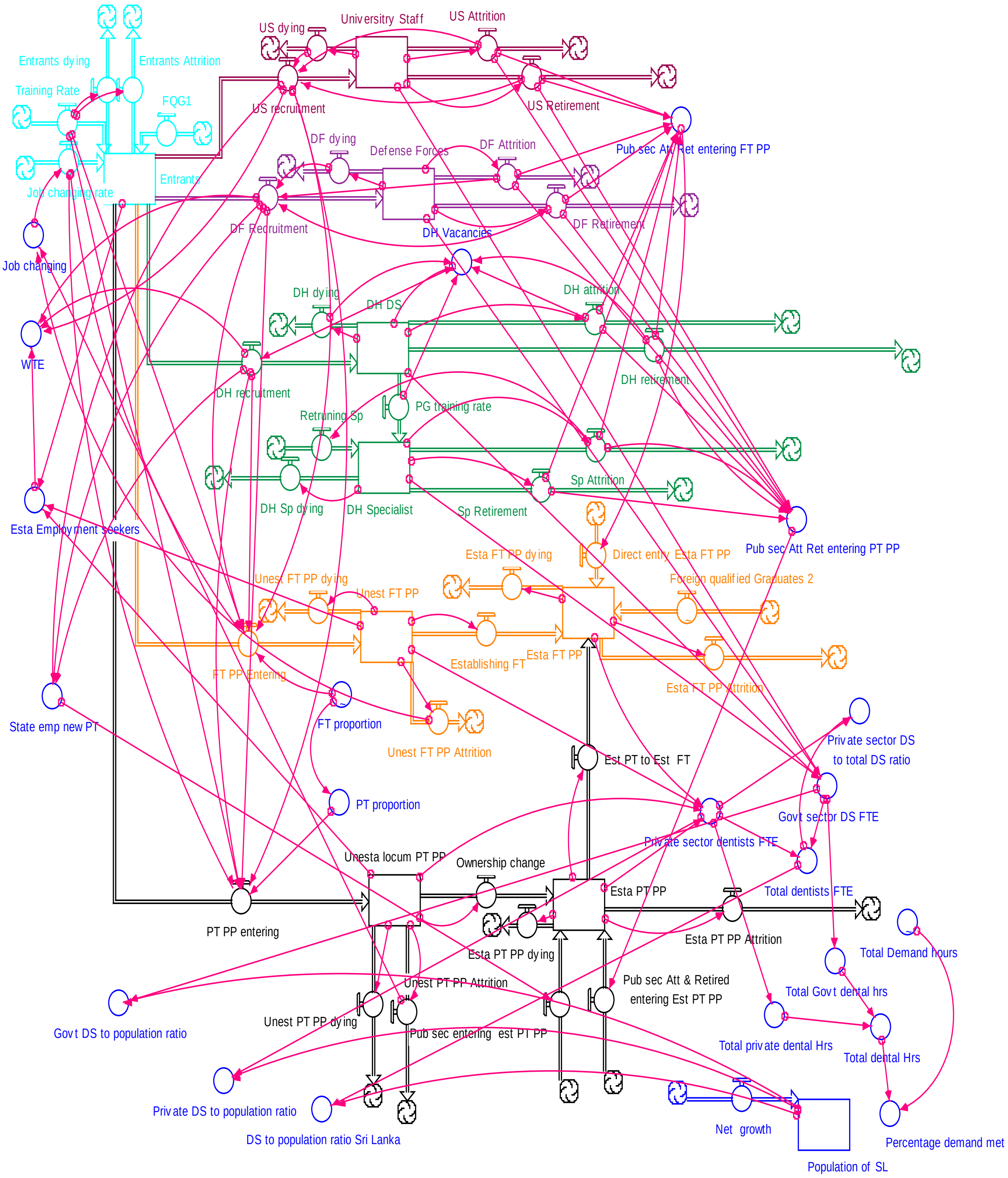
The supply-side model presented in this paper is combined with a demand model (De Silva, 2012), not described in detail here, which enables users to investigate the mismatch between supply and demand in a number of different scenarios. In these scenarios, Sri Lanka's economic development and population growth follow different patterns, each of which is associated with change in demand. For example as people become wealthier the market for cosmetic, preventative and restorative dentistry will grow, whereas in poorer scenarios there would be a greater need for more basic treatments such as extractions. To calculate demand in terms of treatment hours required (and hence into posts), a survey entitled the 'Sri Lanka Timings Inquiry' was conducted to estimate the time taken for various dental treatments in both state and private clinics. This study was similar to the 'Heathrow Timings Inquiry' conducted by the British Dental Association in 1999 (Bearne and Kravitz, 2000), although the range of treatments provided in Sri Lanka is far more limited. The study was conducted in 30 government hospital dental clinics and 20 private dental clinics, covering the entire country. The results were used to determine the total treatment hours required for the whole population and were part of the input data for the demand-side model. Users can also alter the training rate (university intake) and the number of state-sector posts.

Together, the supply-demand model enables policymakers, civil servants and politicians to investigate a range of scenarios and test various staffing, training and recruitment policies. As a practicing dentist, De Silva was able to gather reliable data that was acceptable to the dental fraternity, a crucial part of gaining professional acceptance of the model. He presented the model at a Cabinet subcommittee meeting of the Sri Lankan government and demonstrated the effect of different policy options. Based on the findings, in October 2010 the Sri Lankan Ministry of Higher Education finally agreed to fix the intake of dentistry students at its current level of 80 for another ten years. Moreover, the Ministry of Health was convinced, based on the model results, of the long-term adverse consequences of widespread unemployment among dental surgeons and created 400 additional state vacancies over three years from 2011 to 2014. All these posts were created in rural areas to narrow the urban-rural divide.

The Sri Lankan government has a tool to aid decision-making that takes a 'whole system' view and reduces the effects of political lobbying and game playing by different government ministries with competing objectives. The use of taxpayers' money has been improved and Sri Lankan dental surgeons have improved career prospects. However the main beneficiaries are the Sri Lankan people. By the end of 2012, when only 250 of the planned 400 new posts had been created, more than one million people who previously had no access to care now visit a state dentist at least once a year, and will benefit from improved dental health in future. Moreover, many of these 250 new dentists also work in the private sector after hours, which have resulted in 500,000 additional patient visits per year in the private sector.

De Silva is now Head of Dental Human Resources Planning and Training at the Sri Lankan Ministry of Health. The Sri Lankan Ministry for Health has recently asked him to develop a similar model for midwives and other paramedical staff. He is now a keen advocate of system dynamics modelling for workforce planning and is Sri Lanka's acknowledged expert in this field. He was shortlisted for the IFORS Development Prize in 2011.

APPENDIX: SCREEN PRINT OF THE STELLA MODEL (SUPPLY SIDE ONLY)



REFERENCES

Bearne A and Kravitz A (2000). The 1999 BDA Heathrow Timings Inquiry. *British Dental Journal*, **118(4):** 189-194.

Bronkhorst E M (1995). Modelling the Dutch Dental Health Care System - a comprehensive system dynamics approach. University of Nijmegen, The Netherlands: Trikon.

Central Bank of Sri Lanka (2009). Annual Report. Colombo: Central Bank of Sri Lanka.
De Silva M D K (2012). *Dental workforce planning for Sri Lanka.* PhD thesis, School of Management, University of Southampton, UK.
Edwards J S (1983). A survey of manpower-planning models and their application. *Journal of the Operational Research Society* **34(11)**: 1031-40.
Gass S I (1991). Military manpower-planning models. *Computers and Operations Research* **18(1):** 65-73.
Grossler A and Zock A (2010). Supporting long-term workforce planning with a dynamic aging chain model: a case study from the service industry. *Human Resource Management* **49(5):** 829-48.

AUTHOR BIOGRAPHIES

SALLY BRAILSFORD is Professor of Management Science at the University of Southampton, UK. She received a BSc in Mathematics from the University of London, and MSc and PhD in Operational Research from the University of Southampton. Her research interests include simulation modeling methodologies, system dynamics, health service research and disease modelling, and the modelling of human behaviour in healthcare systems. She is Vice-President 1 of EURO (the European Association of OR Societies), chair of the EURO Working Group on OR Applied to Health Services (ORAHS) and is an Editor-in-Chief of *Health Systems*. She is on the editorial boards of *Health Care Management Science* and the *Journal of Simulation*. Her email address is s.c.brailsford@soton.ac.uk.

DILEEP DE SILVA is the consultant in charge of dental workforce planning and training in the Ministry of Health, Sri Lanka. He received a BDS from the University of Peradeniya, Sri Lanka, and MSc and Doctor of Medicine (MD) degrees in Dental Public Health from the University of Colombo. He also holds an MBA from the University of Colombo and a PhD in Management from the University of Southampton. His research interest includes Health Service Research and HR for Health. He is President-elect of the Sri Lanka Dental Association. His email address is dileepdenta@yahoo.com.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

USING SIMULATION TO ASSESS THE EFFICIENCY AND FAIRNESS OF A VARIETY OF ALGORITHMS FOR THE ADVANCED ONLINE BOOKING OF PATIENTS FOR SURGERY

Miss. Marion L. Penn

University of Southampton
University Road,
Southampton
SO17 1BJ
m.penn@soton.ac.uk

Prof. Chris N. Potts

University of Southampton
University Road,
Southampton
SO17 1BJ
C.N.Potts@soton.ac.uk

Prof. Paul R. Harper

Cardiff University

Senghennydd Road,
Cardiff, Wales, UK
CF24 4AG
harper@cardiff.ac.uk

Dr. Stephen Lash

University Hospital Southampton
NHS Foundation Trust
Tremona Road
Southampton
SO16 6YD
Stephen.Lash@uhs.nhs.uk

ABSTRACT

Hospital operating theatres are costly resources and are often needed for patients with long care pathways, so their efficient use is critical to the overall efficiency of the hospital, both financially and in terms of reducing patient waiting times.

Even if only a single surgeon's patients are considered, there are variations in arrival rates, expected and actual operation times, and urgency of surgery to consider, as well as the fairness of the process. For patients the ideal time to book their surgery is while they are talking to the surgeon in an outpatient clinic, however the literature suggests that for online scheduling problems it is most efficient to delay booking as late as possible.

This paper explores and compares a variety of algorithms for booking patients, using a variety of criteria. It takes inspiration from the literature on scheduling in healthcare, scheduling in general and surgeons suggestions.

Keywords: Healthcare, Simulation, Theatre Scheduling

1 INTRODUCTION

Across the world, operating theatres are among the most expensive of hospital resources and are in high demand with surgical groups competing for theatre time in many cases. In the UK, hospitals face increasingly demanding targets for reducing waiting times and avoiding cancellations. Therefore, it is important that hospitals make efficient use of their resources through detailed planning and efficient systems. As Cardoen et al (2010) demonstrate in their literature review there have been a number of studies across the world exploring various aspect of the planning of operating theatre schedules.

In this paper we consider the advanced booking of surgery for elective patients, which involves scheduling individual patients into the available theatre slots. It is desirable to give patients advanced notice of their surgical appointments so bookings are usually done at or soon after the decision to treat them is made. The patients will have varying levels of need for surgery, so more urgent patients will have due dates for treatment which are closer to the time at which the decision to treat them is made. This variation goes from the most emergency patients who must be treated at once to routine patients who must be treated within hospital or government waiting time targets.

This problem is similar to an online job shop scheduling problem with both release dates and due dates, with variability in time remaining until due date as well as arrival and service rates. As patients are not available for booking until the decision has been made to treat them and patients require treatment within a certain time period, based on either clinical urgency or waiting time targets such as those in the UK. There is also a bin packing element to the problem as patient’s operation durations must fit into the available theatre slots.

This paper demonstrates how simulation can be used to assess a variety of scheduling algorithms, suggested by the more general scheduling literature, judging the algorithms based on the number of patients treated within their due dates, and the fairness of the schedules in terms of how quickly patients with different due dates are treated. Following a discussion of the literature the simulation model and algorithms for testing are introduced; finishing with a selection of the results and discussion of their implications.

2 LITERATURE

The literature on more general scheduling problems suggests a variety of scheduling algorithms, which should be considered in our search to improve the process of booking patients for surgery. First in first out (FIFO) has been traditionally used in hospitals and should therefore be considered, although Bowers (2010) doubts its effectiveness when patients have different levels of urgency. The literature on bin packing problems suggests algorithms such as booking into the slot for which the patients expected theatre time is the best fit to the remaining unbooked theatre time (Shi and Ye, 2008). It also suggests booking in decreasing order of expected treatment time (Wong and Lee, 2009), while the literature concerning due dates (Schmidt, 1988) and online scheduling (Pruhs et al., 2004) recommend increasing order of expected treatment time. The literature on due dates (Chen et al., 1998) recommends arranging jobs in increasing due date order before scheduling them. Also, the online scheduling literature recommends delaying the decisions as much as possible (Lu et al., 2003, Anderson and Potts, 2004, Liu et al., 2009, Potts and Strusevich, 2009, Tao et al., 2010 and Liu et al. 2011).

Some of the above could be used to complement each other, for example using a second rule for tie breaks, while others are completely contradictory. Thus, a thorough consideration of a range of algorithms is required.

In the theatre scheduling literature Dexter et al. (2000), Dexter and Traub (2002), Sciomachen et al. (2005) and Van Houdenhaven et al. (2007) all use simulation to assess aspects of booking algorithms. While none of them consider the range of algorithms we will explore below this supports our decision to use simulation modeling to test such algorithms.

3 MODELING

At the start of this study we were approached by the ophthalmology department of a local hospital, enquiring as to whether we could assist them in improving their booking of patients for surgery. So we are working closely with them to understand the problem and the constraints on it, with the aim of developing scheduling algorithms that they can implement. Thus all of the data for the model are based on a case study of that surgeon’s case mix, with the arrival times and urgency of patients in the model sampled from empirical distributions calculated from historical data.

The surgeon considered operates in two different theatre slots within the week, so these are modelled separately. The patients are divided into emergency and routine patients, and within the routine patients further divided into cataract and VR patients as these types of surgery have different due date and duration distributions. The resulting simulation model is shown in Figure 1.

The loop sending some patients back from booking to the queue is included so that if the algorithm under testing indicates that they should not be booked yet, they are sent back to the queue with a delay so that they aren’t reconsidered for booking again that day.

Emergency patients always jump the queue, and time is reserved in each theatre slot to allow for them.

It is assumed that no rescheduling takes place, as allowing rescheduling could mask flaws in the algorithm under consideration.

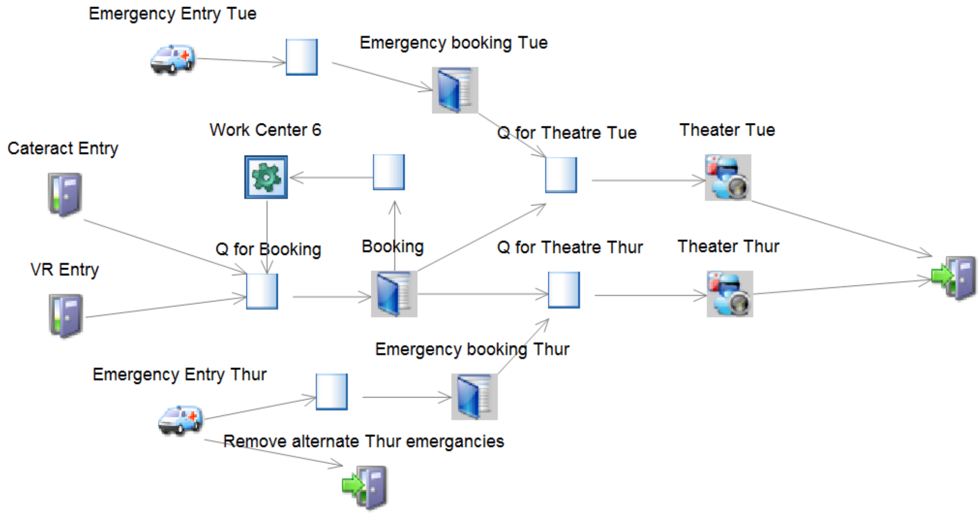


Figure 1 *The simulation model*

The most important values output at the end of each simulation run of 300 weeks or for which average data are returned at the end of each trial with 250 replications are as follows:

- The percentage of each type of patients treated before their due date.
- Separately for cataract and VR patients, the minimum, maximum and mean waiting time for those patients with initial time remaining to target of 4, 10 and 16 weeks.

4 ALGORITHMS

For all of the algorithms patients are considered for booking in order of earliest due date, as investigation (not included here) indicates that this is the best priority order. The names of algorithms are combinations of abbreviations to describe the scheduling policy or policies that they implement as follows:

- ‘F’ – stands for first and indicates that the part of the algorithm discussed is booking patients into the first available slot, effectively FIFO with the arrivals being the order in which patients are considered for booking.
- ‘DD’ – stands for book to due date, this policy was suggested by the surgeon involved and reflects the concern of Bowers (2010) that FIFO fails to allow for varying urgency of patients. It indicates that patients are booked as late as possible within their due dates, or if that is not possible as soon as possible after their due dates.
- ‘nDD’ – early testing revealed that just booking to due date resulted in underutilization of some slots, so it has been adapted to ‘nDD’ which stands for next or due date, indicating that if there is space in the next available slot then the patient should be booked into that space to avoid theatre time being unutilized.
- ‘xw’ – where x is an integer, stands for the number of weeks ahead that booking is permitted by the algorithm, so an algorithm with ‘2w’ will only make the booking if it is within the next 2 weeks. This is inspired by the suggestion from the literature on online scheduling which suggests that scheduling should be delayed as much as possible.
- ‘exact’ – indicates that if the patients expected operation duration is within a small tolerance of exactly fitting the remaining time available for booking in a slot the patient will be booked into that slot. This is inspired by the bin packing literature.

- 'empty' – indicates that the patient will be booked in to the slot with the least bookings of those available within its due date. Also inspired by the bin packing literature.

The second set of algorithms are based on a compromise between giving patients plenty of notice and the suggestion from the online scheduling literature that booking should be delayed as much as possible. They include a proportion of patients who have agreed to be booked at short notice in exchange for a potentially shorter waiting time, along with part of each theatre slot being kept empty until a few weeks ahead of its surgery date. Thus, '0.3on2weekshold15' indicates that 30% of patients will only be booked if the booking is within the next 2 weeks and 15 minutes of each slot will be held back until 2 weeks before the slot takes place.

5 RESULTS

Figure 2 and Figure 3 show the most important results, the key points are the percentage of the most urgent non-emergency patients who are treated by their due dates and whether there is variation between the waiting times for patients with different levels of urgency.

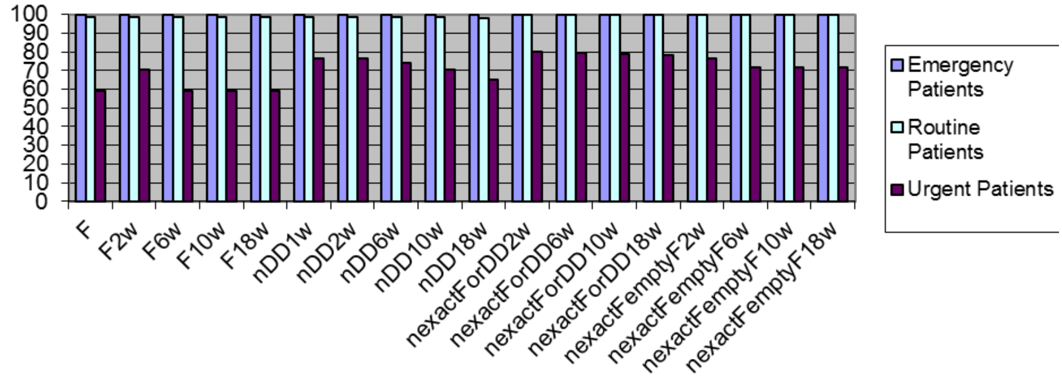


Figure 2 *Percentages of each type of patient seen within their due dates for a various algorithms*

Figure 2 shows that delaying the booking as much as possible does indeed result in better scheduling, particularly for the urgent patients, however this would mean less notice for patients and the difference made varies between algorithms. It also demonstrates that it is worth looking for patients whose operation durations will are 'exact' fits for the remaining theatre space available, which is logical as it makes maximum use of those slots. For patients who don't fill any of the available slots, booking them as close as possible before their due dates leaves space for more urgent patients who arrive after them and this is reflected in the increased proportion of urgent patients treated on time for these algorithms.

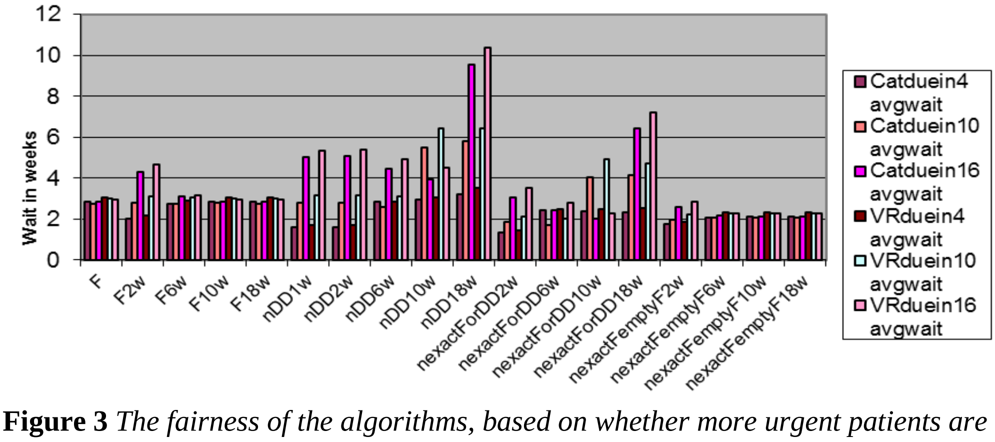


Figure 3 *The fairness of the algorithms, based on whether more urgent patients are have shorter waiting times*

Figure 3 shows that the algorithms where patients are being booked to due date the most urgent patients are experiencing shorter waiting times, indicating that these algorithms are fair to the medical needs of patients. This fits with the concept of fairness that "It is more fair to serve the more urgent jobs, regardless of job sizes or arrival order." (Wiermann, 2011).

For the algorithms in Figure 4 the principle of booking into slots for which the patients expected duration as an 'exact' fit followed by booking as close to their due dates as possible is retained, with the addition of considering only booking a proportion of bookings at shorter notice, with varying amounts of the theatre slots held back for these and the most urgent patients.

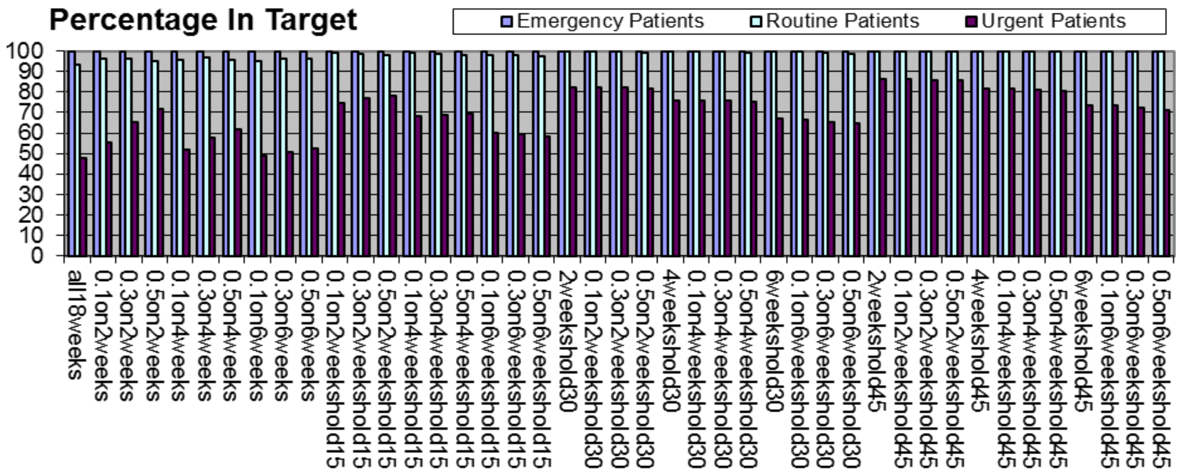


Figure 4 *The percentages of each type of patient seen within their due dates for the algorithms where a proportion of patients are booked at shorter notice*

Figure 4 demonstrates that once part of the theatre slot is held back from booking and only made available for a few weeks before the slot will take place, it does not make much if any difference if some patients are also held back to have their surgery booked at short notice. When part of the slot is held back it is available for the urgent patients when they arrive so it makes sense that this policy is effective at ensuring they can be treated within appropriate time scales whilst allowing the advanced booking of less urgent patients.

6 CONCLUSIONS

The results indicate that booking policies that combine booking patients into slots that they fit 'exactly', booking patients as late as possible within their due dates and holding back part of the theatre slot to be released for booking close to when the slot will take place, will produce the best schedules in terms of treating the most patients within their due dates. The extent and combinations in which they are used will depend on other hospital policies and the relative importance placed on giving patients advanced notice of their theatre dates, avoiding cancellations and fairness. Implementation of these results in the partner hospital is ongoing. This paper also demonstrates the value of simulation modeling for making detailed comparisons of this type.

REFERENCES

Anderson EJ and Potts CN (2004). Online scheduling of a single machine to minimize total weighted completion time. *Mathematics of Operational Research* **29(3)**: 686-697.

Bowers J (2010). Waiting list behaviour and the consequences for NHS targets. *Journal of the Operational Research Society* **61(2)**: 246-254.

Cardoen B, Demeulemeester E and Belien J (2010). Operating room planning and scheduling: a literature review. *European Journal of Operational Research* **201(3)**: 921-932.

Chen B, Potts CN and Woeginger GJ (1998). A Review of Machine Scheduling: Complexity, Algorithms and Approximability. In Du D-Z and Pardalos PM (eds), (1998). *Handbook of Optimization Volume 3*. Kluwer Academic Publishers: Dordrecht, Netherlands. pp.21-171.

Dexter F Macario A and O'Neill L (2000a). Scheduling surgical cases into overflow block time - computer simulation of the effects of scheduling strategies on operating room labour costs. *Anesthesia & Analgesia* **90(4)**: 980-988.

Dexter F and Traub RD (2002). How to schedule elective surgical cases into specific operating rooms to maximise the efficiency of use of operating room time. *Anesthesia and Analgesia* **94(4)**: 933-942.

Liu M, Xu Y, Chu C and Zheng F (2009). Online scheduling to minimize modified total tardiness with an availability constraint. *Theoretical Computer Science* **410(47-49)**: 5039-5046.

Liu Y, Chu C and Wang K (2011). A new heuristic for the operating room scheduling problem. *Computers and Industrial Engineering* **61(3)**: 865-871.

Lu X, Sitters RA and Stougie L (2003). A class of on-line scheduling algorithms to minimize total completion time. *Operations Research Letters* **31(3):** 232-236.

Potts CN and Strusevich VA (2009). Fifty years of scheduling: a survey of milestones. *Journal of the Operational Research Society* **60(S1)**: S41-S68.

Pruhs K, Sgall J and Torng E (2004). Online Scheduling. In. Leung J Y-T (ed). *Handbook of Scheduling: Algorithms, Models and Performance Analysis*. Chapman and Hall/CRC: London. pp. 15-1 - 15-43

Schmidt G (1988). Scheduling independent tasks with deadlines on semi-identical processors. *Journal of the Operational Research Society* **39(3)**: 271-277.

Sciomachen A, Tanfani E and Testi A (2005). Simulation models for optimal schedules of operating theatres. *International Journal of Simulation* **6(12-13)**: 26-34.

Shi Y and Ye D (2008). Online bin packing with arbitrary release times. *Theoretical Computer Science* **390(1)**: 110-119.

Tao J, Chao Z and Tao Y (2010). An optimal semi-online algorithm for a single machine scheduling problem with bounded processing time. *Information Processing Letters* **110(8-9)**: 325-330.

Van Houdenhoven M, Hans EW, Klein J, Wullink G and Kazemier G (2007). A norm utilisation for scarce hospital resources: evidence from operating rooms in a Dutch university hospital. *Journal of Medical Systems* **31(4)**: 231-236.

Wiermann A (2011). Fairness and scheduling in single server queues. *Surveys in OR and Management Science* **16(1)**: 39-48.

Wong L and Lee LS (2009). Heuristic placement routines for two-dimensional bin packing problem. *Journal of Mathematics and Statistics* **5(4)**: 334-341.

AUTHOR BIOGRAPHIES

MARION PENN received a BSc (Hons) Mathematics from the University of Bristol in 2002 and a MSc in Operational Research from the University of Southampton in 2006. She expects to complete her PhD at the University of Southampton in 2014, where she is currently a teaching fellow.

CHRIS POTTS is a professor of Operational Research at The University of Southampton where he is currently head of the Operational Research group. He has published a significant number of papers including various aspects of scheduling.

PAUL HARPER is a professor of Operational Research at Cardiff University where he is currently head of the Operational Research group and a director of the Health Modelling Centre Cymru.

STEPHEN LASH is a Consultant Ophthalmic Surgeon at University Hospital Southampton NHS Foundation Trust. In addition to his medical qualifications he received an MBA from the University of Southampton in 2009.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

THE NEED FOR CLOUD-BASED SIMULATION FROM THE PERSPECTIVE OF SIMULATION PRACTITIONERS

Bhakti S.S Onggo

Department of Management Science
Lancaster University Management School
Lancaster, LA1 4YX, United Kingdom
s.onggo@lancaster.ac.uk

Simon J.E. Taylor

School of Information Systems, Computing and Mathematics, Brunel University
Uxbridge, Middlesex, UB8 3PH, United Kingdom
simon.taylor@brunel.ac.uk

Arman Tulegenov

"InnoForce CA" LLP, office 210, 85A Dostyk street,
Almaty, Republic of Kazakhstan
arman.tulegenov@innoforce.kz

ABSTRACT

Cloud-based simulation (CBS) is one of the new grand challenges in modelling and simulation (M&S). However, the work on web-enabled M&S is not new. A case in point is web-based simulation (WBS). Given the similarities between WBS and CBS, it is important to learn from WBS. Despite advancements in WBS research, its commercial applicability and adoption by users has not grown to the desired extent. This is partly due to the strong emphasis on WBS as a technological tool instead of a socio-technological tool in which users, their needs and circumstances are considered. To understand the needs and perception from practitioners on CBS, we conducted a survey. The results show that practitioners have a good exposure to cloud applications and mobile gadgets. There also appears to be evidence of a need for CBS that provides fast response time, effective communication tools and functionalities to share, store and retrieve models.

Keywords: Cloud-based simulation, Web-based simulation, simulation project, simulation practice, survey

1 INTRODUCTION

Cloud-Based Modelling and Simulation (CBMS) has been mentioned consistently as one of the grand challenges in *Modelling and Simulation* (M&S) (Taylor et al. 2012, Taylor et al. 2013). CBMS is a term used to capture the intersection between M&S and cloud computing. Although the term "cloud computing" has been with us for some time, it is apparent that there is not a universally accepted definition. One of the most popular definitions is provided by Grance and Mell (2011) on behalf of the US *National Institute of Standards and Technology* (NIST). In their definition, cloud computing is "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models." *ibid*. The five essential characteristics of cloud computing are on-demand self-service (automatic deployment of computing capabilities), broad network access using multiple platforms (such mobile devices, laptops and desktop computers), resource pooling, rapid elasticity (computing capabilities can be scaled up and down to match the fluctuation in demand) and measured service (such as pay-as-you-use model). Cloud computing offers three service models: *Software as a Service* (SaaS), *Platform as a Service* (PaaS) and *Infrastructure as a Service* (IaaS). In SaaS, consumer can run applications using cloud infrastructure while in PaaS, consumers can create applications to be run on cloud infrastructure (hence the consumers have control over their applications). In IaaS, consumers can have control not only over their application but also over the storage and the operating systems. Finally, cloud computing can be deployed in four models: public cloud (infrastructure can be used by public), private cloud (infrastructure can only be used by an individual organisation), community cloud (infrastructure can be used by a community of users with shared missions) and hybrid cloud (the combination of the other three deployment models).

Based on the above definition, we can see the relevance of cloud computing and M&S. As is being explored in the *Cloud-based Simulation for Manufacturing and Engineering* (CloudSME) initiative (www.cloudsme.eu), we can imagine the following cloud-based scenarios.

1. Simulation users can run simulation models on cloud infrastructure (*Simulation as a Service*).
2. Some simulation users (especially simulation modellers) may want to have a control over the models by creating models using simulation development tool hosted on cloud infrastructure (*Modelling as a Service*).
3. Some simulation modellers may want to have a greater control over the simulation development tool itself. Provided that the simulation development tool is reconfigurable or decomposable, we can mix and match the components of the tool (for example, we may use a random number generator from one vendor and choose a simulation optimization tool from another vendor). This is close to the PaaS.
4. Simulation users may want to have control over storage (for example, in data-driven simulation) and the execution platform and middleware (for example, in high performance simulation or distributed simulation). This is similar to the IaaS.

The effort to use the Internet or the Web for M&S is not new. The early efforts in utilising the Web to support model design, model execution and analysis of generated simulation results can be traced back to Fishwick's paper on web-based simulation (WBS) in 1996. There was an explosive growth in the number of publications on WBS between 1996 and 2000. After the peak period between 2000 and 2002, the number of publications dropped off very quickly. Given the similarities between the WBS and cloud-based simulation (CBS), it is important to learn from what has happened to WBS.

First, *CBS must not simply re-implement existing simulation software using cloud computing technologies*. This argument is taken from Kuljis and Paul's critique on WBS (Kuljis and Paul 2003). They argue that the problem with the decline of WBS stems from the fact that there was a mismatch between the main characteristics of the Web and the approach taken by the domain of WBS, which failed to take full advantage of the features of the Web including common standards, interoperability, ease of navigation and use, etc. In other words, the focus of many WBS endeavours was on the re-implementation of existing standalone simulation software.

Second, *CBS should be seen as a socio-technical tool instead of a technological tool*. This comes from our observation that there was a strong emphasis on WBS as a technological tool. This view is also reflected in the literature review done by Byrne et al. (2010). WBS should have adequately addressed what the simulation users really needed from its usage in practice.

In this paper, to try to understand what socio-technical underpinnings might exist for CBS, we present the result of a survey that we sent to simulation practitioners. The objective of the survey is to find out how practitioners carry out their simulation project, their exposure to cloud-enabled applications and mobile gadgets, and their perception on a number of functionalities offered by cloud-based simulation. The remainder of this paper is organised as follows. Section 2 explains the survey design and section 3 presents and discusses the result from the survey. Our conclusions are presented in Section 4.

2 SURVEY DESIGN

To find out more about the interest among simulation practitioners on Cloud-based simulation (CBS), we carried out a survey from January 2013 to the end of February 2013. The questionnaire was sent through various mailing lists and LinkedIn groups related to simulation. We also sent the questionnaire to Simul8's and Lanner's customers (both are discrete-event simulation vendors).

The first part of the questionnaire is to identify the field of industry/expertise of the respondents. This is followed by questions related to simulation projects in which they have been involved (team size, frequency of modelling work and role within a project). These questions should allow us to analyse the next questions on the tools they use for their communication with other team members or clients and how often they use communication tools which include: face-to-face meeting, teleconference (voice only), video conference, on-line messaging/chatting and email (they can also specify other tools). The next question asks about the exposure to a number of cloud-enabled applications and mobile gadgets at work and socially. This question should give us a good idea about the familiarity and appreciation of simulation practitioners on cloud-enabled applications and mobile gadgets whether at work or socially. High usage and appreciation may indicate a positive attitude towards cloud-based applications. The last question asks if they have a stable and fast internet connection, how they would appreciate a number of functionalities offered by cloud-based simulation. This question should provide us with what is considered to be essential by simulation practitioners. The detailed questions asked in the questionnaire are given in the appendix.

3 SURVEY ON SIMULATION MODELLING WORK AND TOOLS

We received 86 responses (35 respondents left their contact detail), of which 23% are academics/researchers and the rest come from industry (Figure 1). Figure 1 shows that we have a good cross-section of sectors represented by respondents. Most of them state that they develop a simulation model regularly (Figure 2).

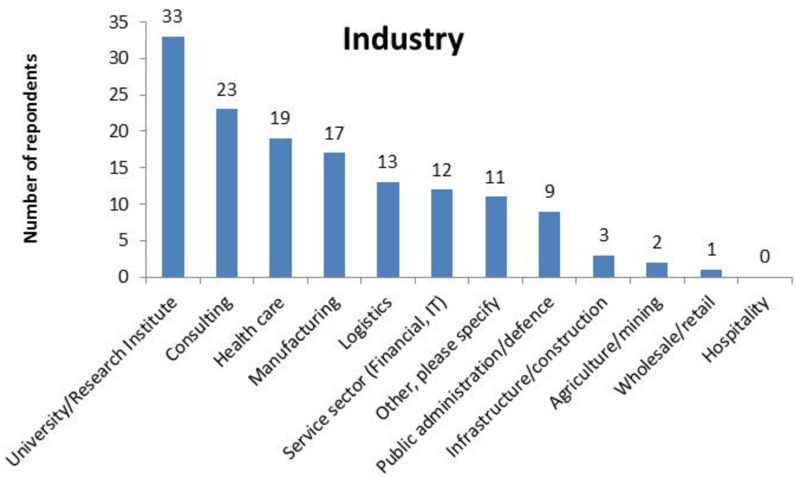


Figure 1 *Sectors represented by respondents*

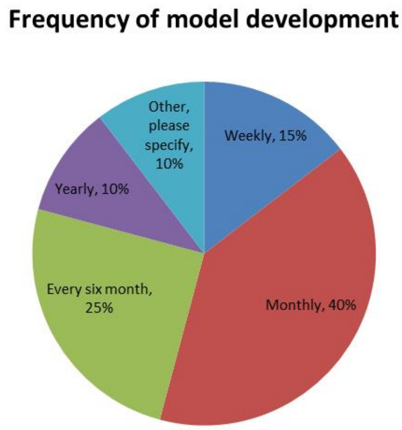


Figure 2 *Frequency of model development*

In the second part of the questionnaire, we tried to find out how in their experience simulation modelling projects had been conducted in practice. Most of them stated that they work in a group and the typical group size is between 3 and 6 (Figure 3). The respondents show that team members in their group have various roles such as simulation modellers, simulation analysts and domain experts (Figure 4). In this survey, we divide the roles into seven categories (the definition of each category is shown in the survey question as shown in appendix). This result confirms that most simulation modelling work requires good communication and collaboration between project team members who may have various roles. The result also shows that face-to-face meetings and emails are the two most common media used for communication between team members (Figure 5). However, the number of respondents who use teleconference and video conference is relatively significant. This may indicate that some team members work from disparate geographical locations. When the communication involves clients who are usually located on a different geographical location, the result shows that email communication is used more than face-to-face meeting (Figure 6). The result also shows a relatively significant number of respondents use teleconference and videoconference facilities to communicate with their clients. However, as one might expect, the majority of interaction is via face-to-face meetings and by email.

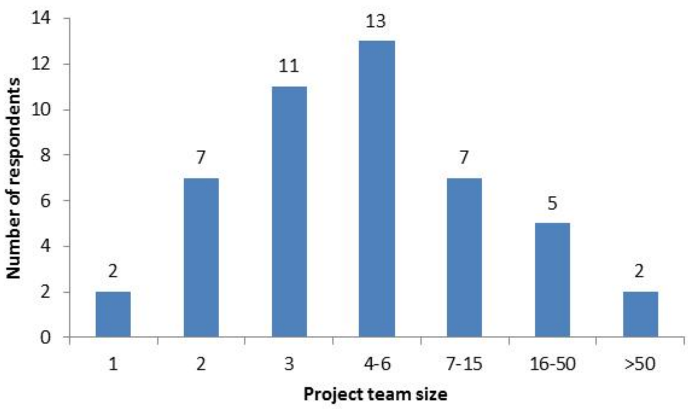


Figure 3 *The typical team size in a simulation project*

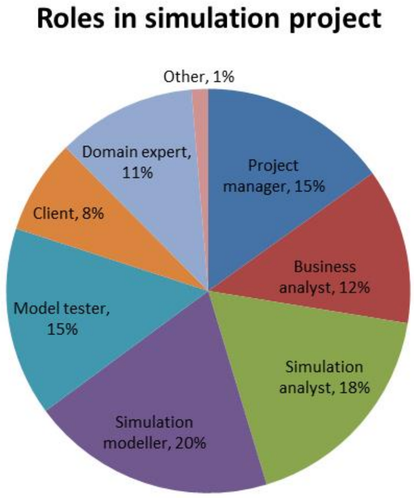


Figure 4 *Roles in simulation project*

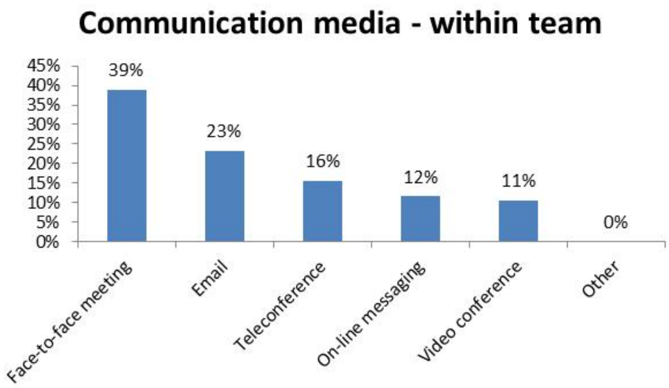


Figure 5 *Communication media used with other team members in a simulation project*

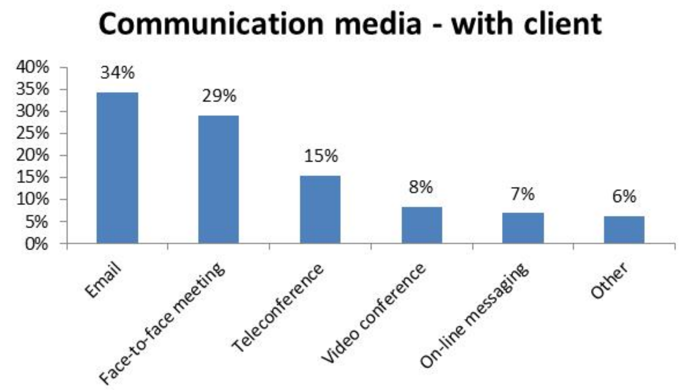


Figure 6 *Communication media used with clients in a simulation project*

The third part of the questionnaire aims to find the exposure of the respondents to popular cloud-enabled/web-enabled applications (such as cloud storage, remote desktop, collaborative writing, social networking and wikis), gadgets (smart phone and tablet), and whether they use the software and the gadgets at work, socially or both. The result is shown in Figure 7. In general, the result shows that the respondents have a good exposure to the tools and gadgets (at work, socially or both). Hence, it implies that many of the respondents have already adopted those technologies. The three most commonly used tools and gadgets at work (including for a simulation project) are remote desktop, cloud storage and a smart phone. This strengthens the previous result that many respondents need to be mobile or to work with people at different physical locations. Other popular tools include collaborative writing (such as Google Docs and Office 365), wikis and blogs. It should also be noted that many respondents have used popular social networking sites (such as LinkedIn and Facebook). It is worth nothing that almost half of those who are using LinkedIn state that they use it for work. Finally, the use of a combination between gadgets (such as smart-phones and tablets) and applications such as cloud storage at work may indicate their readiness to use cloud-enabled applications. In terms of tools specifically used for simulation, it is interesting to note that remote desktop and cloud storage are used the most. Remote desktop allows users to remotely manipulate models between several people. This is highly useful in saving time on projects with avoidable travel.

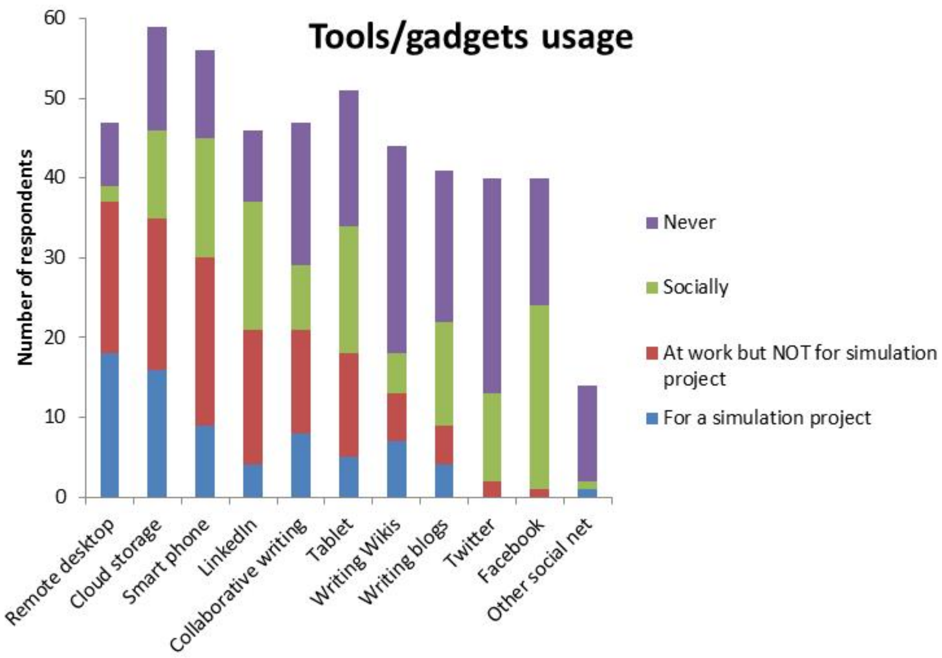


Figure 7 *The exposure and use of tools/gadgets in a simulation project, at work and socially*

The final part of the questionnaire asks the respondents to rate the importance of a number of functionalities offered by cloud technologies to their simulation project, assuming that they have fast and stable internet connection. The result shows that in general most respondents can value the benefit of having those functionalities (Figure 8). It is clear that most respondents value fast response time higher than any functionalities. This is followed by the functionalities that allow other stakeholders in a project team to comment and to run the model using a web browser. This indicates the need of feedback from other stakeholders about the model (from the model developer's side) or giving feedback to those who develop the model. With a web application, it is easier to link a model and the comments on the model and the person who gives the comment may do it at his/her own time from anywhere in the world. Next, the respondents seem to value the functionalities to share, store and retrieve model through a web server or cloud storage relatively highly. The functionalities that are slightly less valued are related to model development process (edit and run models). This is probably due to the familiarity with a desktop tool for model development. This might also be true if we ask whether people who are familiar with writing a document using Microsoft Word (desktop version) to rate the need for the functionality to write a document using Office 365 or Google doc. The need for model repository may be felt more strongly by those who develop model more frequently (60% of respondents who rate this functionality as essential or "good to have" develop model on weekly or monthly basis). Finally, the functionality to allow other stakeholders to edit model is the least valued by most respondents. This may indicate that there is a clear job responsibility in which simulation modellers are the ones who are mostly responsible for model development. Hence, unless there are more than one model developers in a team, this functionality is seen to be the least essential.

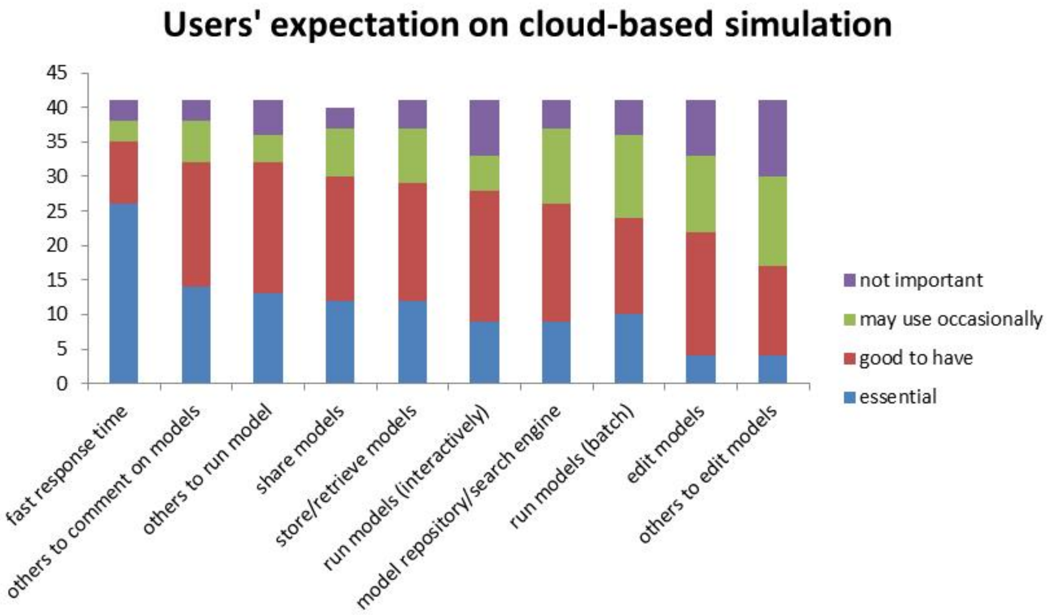


Figure 8 *Users' expectation on cloud-based simulation*

4 CONCLUSION

The findings from this paper contribute to a better understanding of simulation practitioners, their modelling work, their exposure to cloud-enabled applications and mobile gadgets and their appreciation on functionalities that might be offered by cloud-based simulation tools. Given the number of samples, even though it may not represent all simulation practitioners, the findings are considered to be significant. The results confirm that most practitioners work in a team. From the communication tools they use, there is evidence that a significant number of team members work from disparate geographical locations and that most make use of a range of communication technologies to support their work. Simulation practitioners are found to have good exposure to cloud-enabled applications and have used mobile gadgets at work. This shows that they have the potential to be adopters of cloud-based simulation. Simulation practitioners appreciate cloud-based simulation if it can provide fast response time, good communication tools so that feedback from team members/clients can be communicated more effectively and functionalities to share, store and retrieve model through a web server or cloud storage. They seem not to regard functionalities for collaborative model development to be essential although there is evidence of their use. The findings provide us with the foundation to discuss about the roadmap for cloud-based modelling and simulation that is not solely based on technological push but also the real demand from simulation practitioners. This will be addressed in our future work.

A APPENDIX: QUESTIONNAIRE

Q1.1 Please select the most appropriate option(s) which refers to your field of industry/expertise? Agriculture/mining, Consulting, Health care, Hospitality, Infrastructure/construction, Logistics, Manufacturing, Public administration/defence, Service sector (Financial, IT), University/Research Institute, Wholesale/retail, Other, please specify ____________

Please consider a "typical" simulation project that you have been involved in to answer the following questions

Q2.1 What was your project team size (number of people)?

Q2.2 How often do you (or your project teams) create simulation models? Weekly, Monthly, Every six month, Yearly, Other, please specify ________________

Q2.3 What is your main role(s) within a simulation project? Project manager (plan, budget, monitor progress, etc.), Business analyst (interact with clients, understand their needs and translate them into modelling objectives for the simulation analyst), Simulation analyst (analyse simulation input data, simulation output analysis, specify model requirement for simulation modeller), Simulation modeller (develop a model), Model tester (test model and validate model), Domain expert (someone who is an expert on the industry or system being modelled), Client (the beneficiary/user of the model), Other, please specify ____________________

Q2.4 How often did you communicate with other team members using the following media (in percentage)? Face-to-face meeting, Teleconference (voice only), Video conference, On-line messaging/chatting, Email, Other, please specify ____________________

Q2.5 How often did you communicate with your clients (or simulation consultants if you are a client) using the following media (in percentage)? Face-to-face meeting, Teleconference (voice only), Video conference, On-line messaging/chatting, Email, Other, please specify __________________

Q3.1 Do you use the following tools or gadgets?

	For a simulation project	At work but NOT for simulation project	Socially	Never	I don't know
Remote desktop / desktop sharing	❑	❑	❑	❑	❑
Collaborative writing (such as Google Docs and Office 365)	❑	❑	❑	❑	❑
Writing blogs	❑	❑	❑	❑	❑
Writing Wikis	❑	❑	❑	❑	❑
LinkedIn	❑	❑	❑	❑	❑
Facebook	❑	❑	❑	❑	❑
Twitter	❑	❑	❑	❑	❑
Other social networking application, please specify	❑	❑	❑	❑	❑
Cloud storage (such as DropBox, SkyDrive, GoogleDrive)	❑	❑	❑	❑	❑
Tablet (such as iPad, Galaxy Tab and Google Nexus)	❑	❑	❑	❑	❑
Smart phone (such as iPhone, Samsung Galaxy)	❑	❑	❑	❑	❑

Q3.2 If you have fast and stable internet connection, how do you rate the following functionalities in terms of their importance to your simulation project?

	essential	good to have	may use occasionally	not important
store/retrieve models on a web server / cloud	❍	❍	❍	❍
run models and obtain results using a web browser	❍	❍	❍	❍
see the animation of simulation runs using a web browser	❍	❍	❍	❍
edit models using a web browser	❍	❍	❍	❍
share models through a web server / cloud	❍	❍	❍	❍
allow relevant stakeholders to run models using a web-browser	❍	❍	❍	❍
allow relevant stakeholders to edit models using a web browser	❍	❍	❍	❍
allow relevant stakeholders to comment on models using a web-browser	❍	❍	❍	❍
fast response time (animation runs smoothly, produce simulation results without delay, responsive editting, etc.)	❍	❍	❍	❍
allow the creation of a model repository with specialised search engine to find suitable models for reuse	❍	❍	❍	❍

REFERENCES

Byrne J, Heavey C and Byrne P J (2010). A review of Web-based simulation and supporting tools. *Simulation Modelling Practice and Theory* **18**: 253-276.

Fishwick P A (1996). Web-Based Simulation: Some Personal Observations. In: Charnes J M, Morrice D J, Brunner D T and Swain J J (eds). *Proceedings of the 28th Winter Simulation Conference*. IEEE Computer Society: Piscataway, NJ, USA, pp 772–779.

Grance T and Mell P (2011). The NIST Definition of Cloud Computing. NIST [Online]. Available at: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf [Accessed 23/12/12].

Kuljis J and Paul R J (2003). Web-based discrete event simulation models: Current states and possible futures. *Simulation & Gaming* **34(1)**: 39-53.

Taylor S J E, Fishwick P A, Fujimoto R, Page E H, Uhrmacher A M, Wainer G (2012). Panel on Grand Challenges for Modeling and Simulation. In: Laroque C, Himmelspach J, Pasupathy R, Rose O and Uhrmacher A M (eds). *Proceedings of the 2012 Winter Simulation Conference*. IEEE Computer Society: Piscataway, NJ, USA.

Taylor S J E, Khan A, Morse K, Tolk A, Yilmaz L and Zander J (2013). Grand Challenges on the Theory of Modeling and Simulation. In: Proceedings of the Symposium on Theory of Modeling and Simulation, 9 pages.

AUTHOR BIOGRAPHIES

BHAKTI S S ONGGO is a lecturer in the Department of Management Science at the Lancaster University Management School, Lancaster, United Kingdom. His research interests lie in the areas of simulation methodology (modelling paradigms and conceptual modelling), simulation technology (parallel and distributed simulation, cloud-based simulation), business process modelling and simulation applications. His email address is s.onggo@lancaster.ac.uk and his web address is www.lancaster.ac.uk/staff/onggo.

SIMON J E TAYLOR is the convener of this new phase of Grand Challenge activities. He is the Founder and Chair of the COTS Simulation Package Interoperability Standards Group under SISO. He is the Editor-in-Chief of the UK Operational Research Society's (ORS) Journal of Simulation and the Simulation Workshop series. He was Chair of ACM's SIGSIM (2005-2008) and is a member of the SIGSIM Steering Committee. He is a Reader in the Department of Information Systems and Computing and Brunel and leads the ICT Innovation Group. He has published over 150 articles in modelling and simulation. His recent work has focused on the development of standards for distributed simulation and grid- and cloud-based simulation in industry (www.cloudsme.eu) as well as the spread of the international Grid Infrastructure into Africa (www.eI4Africa.eu). His email address is simon.taylor@brunel.ac.uk and his web address is www.brunel.ac.uk/~csstsjt.

ARMAN TULEGENOV is a researcher in "InnoForce CA" LLP, Almaty, Republic of Kazakhstan. His day-to-day operations are focused on implementation of simulation approaches in solutions for companies in fast growing markets. His email address is arman.tulegenov@innoforce.kz.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

CLOUDSME: DEVELOPING A CLOUD COMPUTING-BASED PLATFORM FOR SIMULATION IN MANUFACTURING AND ENGINEERING

Dr. Simon J. E. Taylor

Department of Information Systems and Computing
Brunel University
Uxbridge, Middlesex, UB8 3PH, United Kingdom
simon.taylor@brunel.ac.uk

Dr. Tamas Kiss

Centre for Parallel Computing
Department of Business Information Systems
University of Westminster
115 New Cavendish St, London, W1W 6UW, UK
t.kiss@westminster.ac.uk

Prof. Peter Kacsuk

MTA SZTAKI
1132 Budapest
Victor Hugo u. 18-22
Hungary
kacsuk.peter@sztaki.mta.hu

Prof. Gabor Terstyanszky

Centre for Parallel Computing
Department of Computer Science and Software Engineering
Faculty of Science and Technology
University of Westminster
London, W1W 6UW, United Kingdom
G.Z.Terstyanszky@westminster.ac.uk

Mr. Nicola Fantini

ScaleTools Schweiz AG (Headquarter)
Huobstrasse 10
8808 Pfäffikon
Switzerland
Nicola.Fantini@scaletools.com

ABSTRACT

There is a demand from simulation users for speed, the ability to obtain quickly results from simulation runs. Cloud computing offers the potential for scalable, on-demand access to resources that can be used to speed up simulation. However, developing cloud computing solutions for industry is difficult without appropriate expertise. This paper introduces the CloudSME project that is dedicated to developing cloud computing solutions for simulation in industry and presents an overview of the CloudSME Simulation Platform that is being used to support an initial set of Cloud-based versions of Simul8's discrete-event simulation environment, Ascomp's TransAT computational fluid dynamics application, Ingecon's 3D Scan Insole Designer tool and 2MORO's Bfly software for aircraft maintenance logistics. The paper presents an overview of cloud computing, the CloudSME project, the architecture of the CloudSME Simulation Platform and its two key elements: gUse and CloudBroker.

Keywords: Cloud computing, simulation, discrete-event simulation, computational fluid dynamics, 3D modelling, aircraft logistics.

1 INTRODUCTION

In a recent survey of what practitioners need from web-based simulation, over half indicated that fast simulation response time, as shorter time as possible to obtain results from simulation runs, was the most desirable feature (Onggo, et al. 2014). Cloud computing is rapidly becoming a significant field (Velte, et al. 2009). According to the US *National Institute of Standards and Technology* (NIST), cloud computing is “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction (Grance and Mell 2011).” In other words cloud computing can speed up your application, in theory at least.

To what extent can cloud computing deliver simulation results faster? What additional benefits can “the cloud” bring to modelling and simulation (M&S)? The effective use of cloud computing for high performance computing has been consistently identified as a M&S Grand Challenge (Taylor, et al. 2012, Taylor, et al. 2013a, Taylor, et al. 2013b, Taylor et al. 2013c). To investigate this and to develop and deliver high performance simulation tools for manufacturing and engineering, the CloudSME consortium (www.cloudsme.eu) has brought together researchers and practitioners from the worlds of M&S, high performance computing and cloud service provision. This paper reports on the current state of development of a novel set of cloud-based high performance simulation tools that include Simul8’s discrete-event simulation environment (www.simul8.com), Ascomp’s TransAT computational fluid dynamics application (http://www.ascomp.ch/transat/), Ingecon’s 3D Scan Insole Designer tool and 2MORO’s Bfly software for aircraft maintenance logistics. The paper is structured as follows. In section 2 we review aspects of cloud computing. Section 3 introduces the aims and objectives of the CloudSME project. Section 4 describes the key features of the CloudSME Simulation Platform that is being used to implement cloud versions of the above software. Section 5 concludes the paper.

2 CLOUD COMPUTING

As outlined above, cloud computing can deliver flexible on-demand computing resources. In their NIST report, Grance and Mell (*ibid*) go on to identify a generic cloud model that consists of five essential characteristics:

- on-demand self-service (automatic deployment of computing capabilities),
- broad network access using multiple platforms (such mobile devices, laptops and desktop computers),
- resource pooling,
- rapid elasticity (computing capabilities can be scaled up and down to match the fluctuation in demand) and
- measured service (such as pay-as-you-use model).

Three service models appear to be generally agreed. These are:

- *Software as a Service* (SaaS),
- *Platform as a Service* (PaaS) and
- *Infrastructure as a Service* (IaaS).

SaaS translates a software application so that it runs on a cloud, it is accessible as a cloud-based service that allows consumers to access and run the application on demand. PaaS provides the facilities (the platform) to create and manage these application-based services. IaaS provides the overall components to build, manage and deliver a complete cloud-based system consisting of network access, computing, storage and appropriate middleware.

Four service provision models are also possible:

- *Private clouds* are in-house or hosted for internal use of a particular organization only. They are multi-tenant across organization sites, departments, groups and/or users, and typically mainly focus on self-service and accountability inside the organization.

- *Community clouds* are not only for a single organization, but for a larger, but still separate community across different organizations and/or individuals. They follow similar principles, but more emphasize the sharing aspects of cloud.
- *Public clouds* are offered by certain organizations or cloud providers to a larger community or basically to everybody. They are multi-tenant across organizations and/or individuals, and focus usually on the on-demand and pay-per-use advantages of cloud computing.
- *Hybrid clouds* mix public, community and/or private clouds. They typically focus on providing scalability and failover features.

Building on these, Tolk discusses the concept of using cloud computing to deliver composable M&S services, i.e. Modelling and Simulation as a Service (MSaaS) (Taylor, et al. 2013a). The notion is attractive. If a "typical" M&S application consists of facilities to develop, simulate and visualise a model then one might wish to deploy the application on to a cloud to make it widely accessible via a web-based portal. Model development and visualisation facilities would reside in the portal and would allow users to transparently run simulation experiments on the cloud. The speed at which these experiments would run would be dependent on the amount of money that the user is prepared to spend (up to a practical limit).

There has been much interest in research that could be classified as MSaaS. However, as reviewed by Zhao, et al. (2012) and Sakellari and Loukas (2013), most of this work has been to use M&S to support cloud research rather than using cloud computing *for* M&S. There are a relatively small number of examples where MSaaS is specifically studied. For example, Rossetti and Chen (2012) investigate a cloud computing architecture for supply chain network simulation, Hong and Luo (2011) discuss the use of cloud computing to support large scale ranking and selection, and Lu et al. (2012) investigate architectures for cloud-based simulation. In industry, for example, Simul8 (www.simul8.com) have cloud-based versions of their software and is essentially a web-based simulation that allows users to conveniently access and share models via a cloud. Overall cloud computing offers great opportunities for MSaaS in terms of speeding up experimentation and testing, allowing more experimentation and testing during a simulation project, better accuracy of results and wider use of optimisation techniques made possible by increased experimentation. However, there is a research and development gap that needs to be addressed. The next section introduces CloudSME, a major new project that is developing commercial cloud computing products for M&S in manufacturing and engineering.

3 CLOUDSME

The Cloud-based Simulation platform for Manufacturing and Engineering (CloudSME) project (www.cloudsme.eu) aims to develop a cloud-based simulation platform that enables SME end-users, especially from manufacturing and engineering, to get access to simulation services via a "one-stop-shop." It is also intended that the platform will provide access to cloud computing-based and other distributed computing infrastructures (DCIs) (e.g. desktop grid, HPC cluster, grid, etc.) to support the simulation experimentation.

CloudSME intends to create a combination of PaaS and SaaS solutions to support MSaaS, i.e. a PaaS solution will allow simulation software providers or consultant companies to build SaaS solutions for end-users and deliver MSaaS. An important feature of this is that given the complexity of cloud-based solutions, the access mechanism to the simulation software should completely hide the complexity and potential heterogeneity of the cloud platform and should allow the simulation software to utilise different types of clouds based on their service provisioning models (private, public, community or hybrid cloud), and the underlying cloud middleware. The CloudSME project includes cloud platform providers (CloudBroker (Switzerland) and SZTAKI (Hungary)). The already existing and already integrated CloudBroker (www.cloudbroker.com) and WS-PGRADE platform (SZTAKI) (Kacsuk et al. 2012) will serve as the basis for the PaaS solution, and will enable the simulation software to access HPC cloud resources in a user-transparent way.

There are two types of end users involved in this project: simulation software providers and end users who either use simulation to improve their business or use simulation software as part of their business offering. Simulation software providers currently include Simul8 (UK), ASCOMP

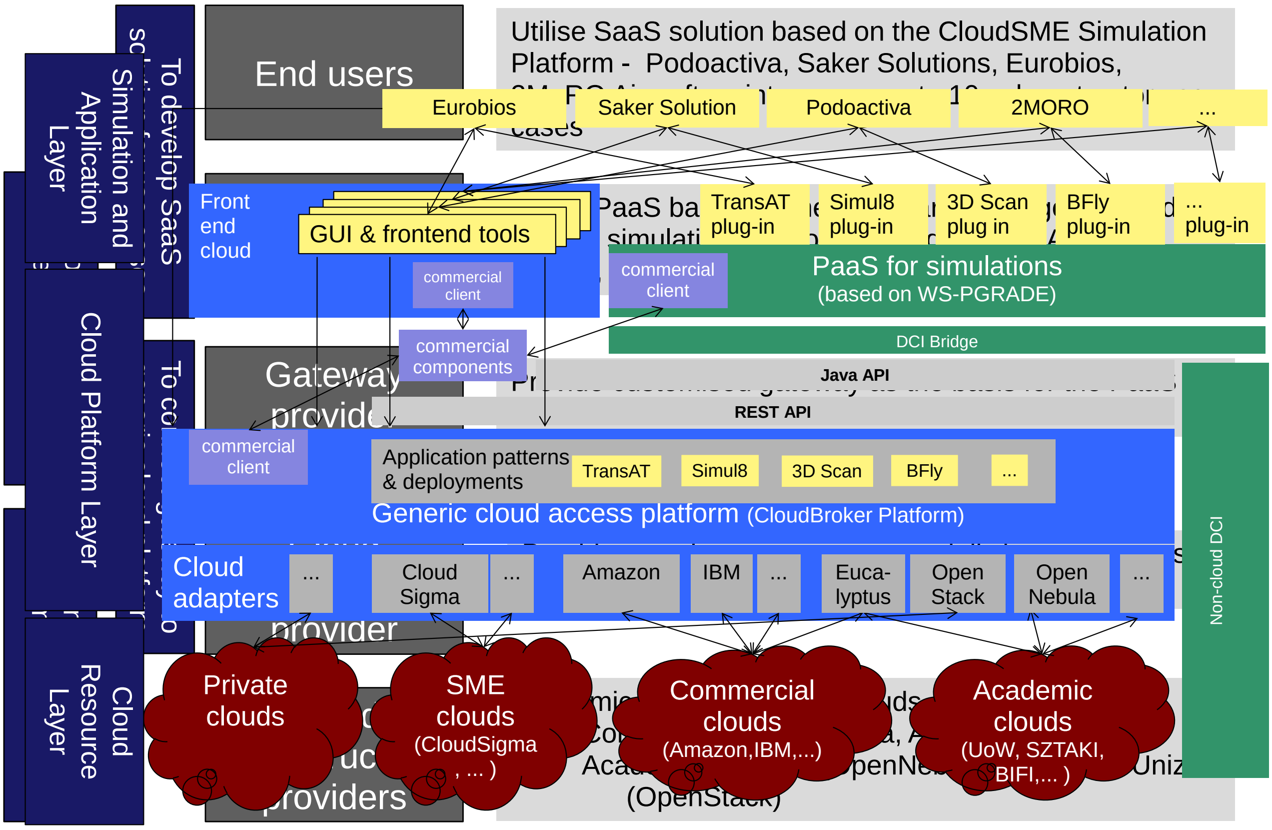


Figure 2 *CloudSME Simulation Platform*

(Switzerland), INGECON (Spain) and 2MoRO (France). End users currently include Saker Solutions (UK), Podoactiva (Spain), Eurobios (France) and Cutting Tools (UK).

Figure 1 shows the key elements of the project. The Centre for Parallel Computing at the University of Westminster (UK) manages the project. Brunel University (UK) has the responsibility for managing the development of the cloud-based products and end user adoption. CloudSigma (Switzerland), SZTAKI, University of Westminster (UK) and UNIZAR (Spain) support the project with cloud resources.

The main objectives of the project are therefore to

- Develop a Platform as a Service (PaaS) solution for simulation software provider companies and their clients in manufacturing and engineering
- Create a generic simulation platform for engineering and manufacturing (the CloudSME Simulation Platform) that supports multiple simulation software.
- Enable simulation software providers to offer cloud-based SaaS simulation solutions for SME end-users
- Enable SMEs in the manufacturing and engineering domain to access simulation services that significantly contribute to improve the efficiency of their operations and/or the quality of their products
- Provide seamless access to High Performance Cloud resources in order to speed up the simulations on-demand

We now discuss the CloudSME Simulation Platform.

4 THE CLOUDSME SIMULATION PLATFORM

The generic structure of the CLOUDSME Simulation Platform (CSSP) is illustrated in Figure 2. The CSSP is divided into the Cloud Resource Layer, the Cloud Platform Layer and the Simulation and Application Layer.

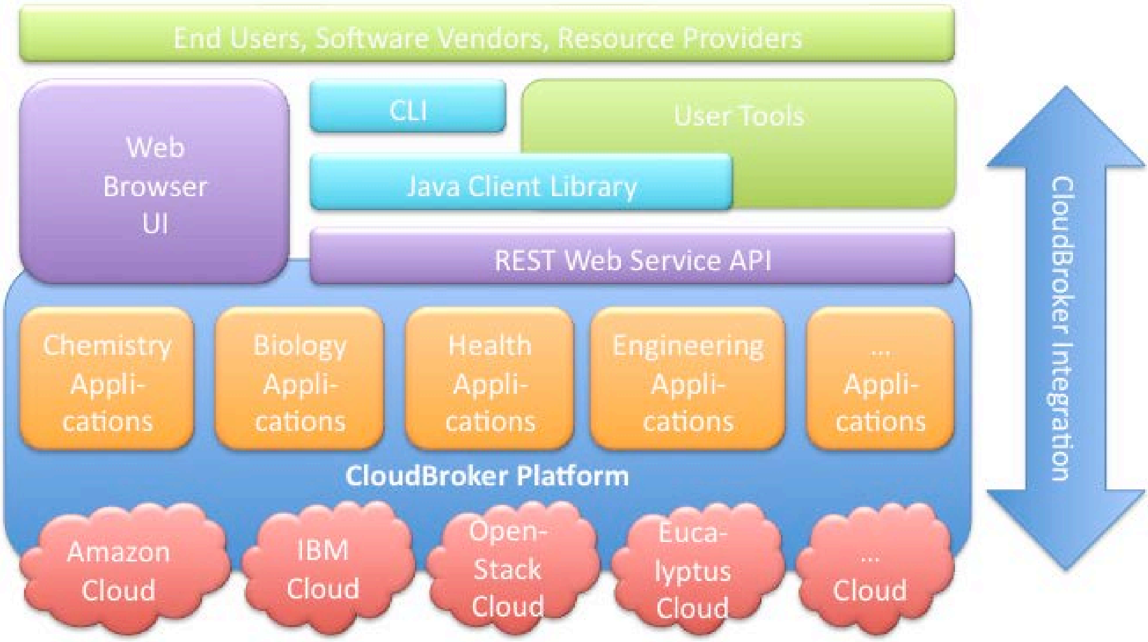


Figure 3 *Architecture of the CloudBroker Platform*

4.1 Cloud Resource Layer

These resources represent the state-of-the-art of cloud technology and consist of private clouds, SME clouds, commercial clouds and academic clouds. In the project these cloud resources are provided by one commercial company (CloudSigma running KVM/Qemu Custom Proprietary stack) and three academic institutes (SZTAKI running OpenNebula, University of Westminster and UNIZAR both running OpenStack). Commercial clouds are also accessible in the layer (e.g. from Amazon EC2 or IBM SmartCloud Enterprise). These heterogeneous distributed resources are transparently connected to the Cloud Plaform Layer via a series of Cloud Adaptors. This way the CSSP provides completely transparent access to these resources for the end-users.

4.2 Cloud Platform Layer

The Cloud Platform Layer will be built of top of the existing CloudBroker Platform and WS-PGRADE/gUSE framework. This layer is crucial in providing a generic basis on which SaaS cloud solutions for manufacturing/engineering SMEs can be built. Each technology is now described.

4.2.1 The CloudBroker Platform

The CloudBroker (CB) Platform is a web-based application store for the deployment and execution of compute-intensive scientific and technical software in the cloud. It is provided as a cloud service that allows for on-demand, scalable and pay-per-use access via the Internet. CB currently offers the platform as public version under https://platform.cloudbroker.com, as hosted or in-house setup, as well as licensed software. This also means that the platform can be run at different physical places and under different legislation if desired. A general overview of the platform architecture is shown in figure 3.

CB uses IaaS from resource providers and provides PaaS to software vendors and SaaS to end users. It offers a marketplace where users can register, deploy, charge for and use their own cloud resources and application software or use resources and software provided by others (e.g., CB itself). Surcharges for platform usage are derived as percentage of the resource and software prices. From this follows a freemium model, that is, if resource providers and software vendors set zero prices for their resources and software, the corresponding platform usage is also for free.

CB incorporates adapters both to public and private cloud infrastructures, and both to compute and storage resources. Currently supported technologies include:

- Amazon EC2 compute and S3 storage resources
- IBM SmartCloud Enterprise compute and Nirvanix storage resources
- OpenStack compute resources (storage in preparation)
- Eucalyptus compute and storage resources
- OpenNebula compute resources (in preparation)

On the application side, CB is cross-domain and supports all kinds of non-interactive, batch-oriented applications, both serial and parallel ones, with a focus on software running under Linux. Windows is also supported. CB has already ported various software in different scientific and technical fields to the platform, in particular in chemistry, biology, health and engineering (see www.sci-bus.eu).

CB can be accessed in several ways. Its main two operation modes to manage and use software in the cloud are either as direct front-end or as back-end middleware service. For the former, the platform can be accessed with any regular web browser as a service via the internet. This is fine for first-time and individual usage. However, for frequent, advanced and automatic usage, API access is provided. These include REST web service interface, Java client library and Linux shell command line interface (CLI). Via its different APIs, CB can be utilized by front-end software as middleware to allow access to applications in the cloud.

CB has been integrated with the WS-PGRADE/gUSE framework (see later) within the FP7 SCI-BUS project ("SCIentific gateway Based User Support" – www.sci-bus.eu). SCI-BUS aims at making scientific gateways available that can transparently utilize different distributed computing infrastructures. The corresponding web portals provide workflows using the Liferay and WS-PGRADE/gUSE frameworks and can access applications on commercial and private cloud infrastructures using the CB Platform.

The current CloudBroker Platform incorporates two main types of security features:

- communication layer security using SSL transport layer encryption both between client and platform and between platform and cloud infrastructures
- authentication and authorization security using login and password for each user and different levels of organization and user roles

The platform employs industry standard application and server technology and is typically operated in industry standard secure data centres. The different cloud infrastructures it utilizes usually also provide authentication mechanisms, isolated virtual machines and security-certified cloud technologies and data centres.

4.2.2 The WS-PGRADE/gUSE Framework

WS-PGRADE/gUSE is a gateway framework that enables the fast and easy creation of application-specific gateways for various user communities and companies. The framework is shown in Figure 3. WS-PGRADE is the web-based user interface (or front-end) while gUSE is a web service-based back end that provides a set of high-level services to access a large set of Cloud and Grid resources. WS-PGRADE is based on Liferay and hence can be easily extended with various portlets either from the Liferay Portlet Repository or by community-developed or company-developed new portlets. It also provides a graphical graph editor to develop WS-PGRADE scientific workflows. gUSE services process WS-PGRADE scientific workflows and control the submission of workflow tasks. gUSE can be used directly via the ASM API (for portal development) or the Remote API (to interface directly with a software package). The framework connects to the following DCIs via the DCI bridge:

- Cluster (PBS, LSF)
- Supercomputers
- Service Grids (ARC, gLite, Globus GT2, GT4, GT5, UNICORE)
- Desktop Grids (BOINC, SZDG)
- Clouds (via CloudBroker Platform: Amazon, IBM, Openstack, OpenNebula)

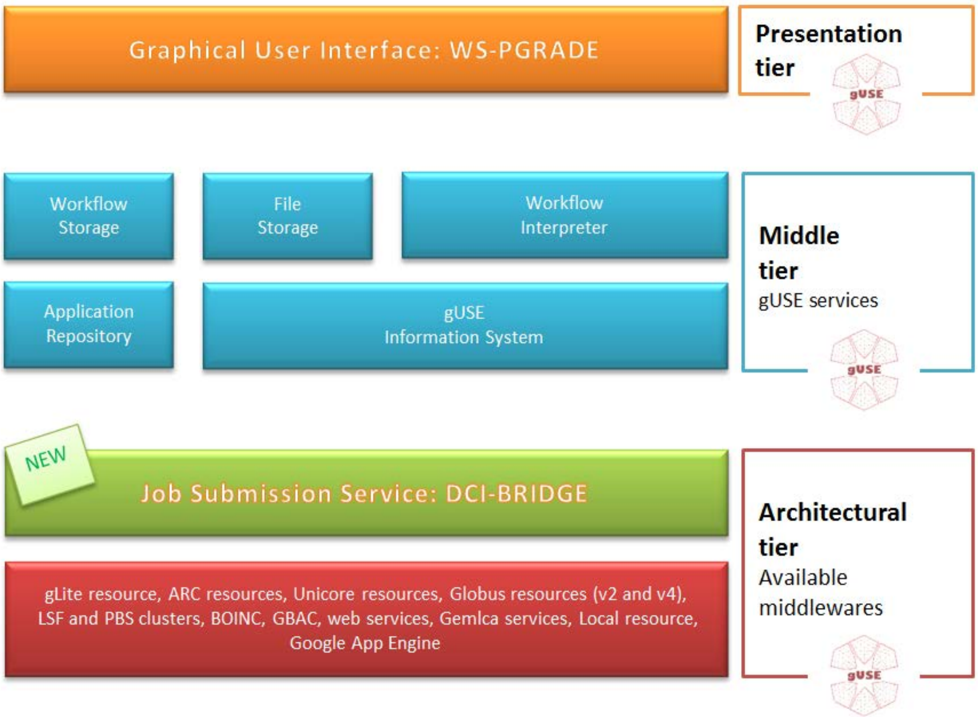


Figure 3 *WS-PGRADE/gUSE Framework*

WS-PGRADE/gUSE has been developed by MTA SZTAKI as an open source software that is available in Sourceforge at: http://sourceforge.net/projects/guse/. The WS-PGRADE/gUSE framework is very popular among the different user communities that is proven by the fact that so far more than 5.000 downloads from more than 40 countries have been done from Sourceforge.

WS-PGRADE/gUSE is actively used by many different user communities and NGIs both in Europe and outside Europe. In the SCI-BUS project 13 different user communities and 4 companies are developing their own gateway based on WS-PGRADE/gUSE.

To use this framework, users develop their applications as workflows using the WS-PGRADE graphical interface. In a simple workflow that could support simulation, for example, a three node workflow would be created. These would be linked together as generator -> parameter scan -> collector. The generator node would have an interface from which a user could upload jobs. These would be passed to the parameter scan node that would distribute the work via gUSE to selected DCIs via the DCI-BRIDGE (in this case CloudBroker would then manage the submission of the work to the respective cloud). When finished results would be passed back and collected for the use in the collector node.

4.3 Simulation and Application Layer

The highest layer of the CloudSME Simulation Platform consists of the simulation software solutions that will be ported to and customised according to the requirements of the cloud-based provision, and the actual SaaS based solutions directly utilisable by the end-users. Currently this includes Simul8's discrete-event simulation environment (www.simul8.com), Ascomp's TransAT computational fluid dynamics application (http://www.ascomp.ch/transat/), Ingecon's 3D Scan Insole Designer tool and 2MORO's Bfly software for aircraft maintenance logistics.

5 CONCLUSION

Cloud computing can bring widespread benefit to simulation in industry by making simulation software more widely accessible and by speeding up the time taken to receive results from simulation runs. Currently the four simulation software that have been identified in this paper are being ported to the CloudSME Simulation Platform. The platform has already been successfully used to implement cloud-based scientific software. The results of developing these four cloud-based industrial applications will be reported in a later paper.

ACKNOWLEDGMENTS

This work is part funded by the CloudSME— Cloud-based Simulation platform for Manufacturing and Engineering project No. 608886 (FP7-2013-NMP-ICT-FOF). The authors kindly acknowledge the efforts of the other beneficiaries on the project.

REFERENCES

Liu X, Qiu X, Chen B and Huang K (2012). Cloud-Based Simulation: The State-of-the-Art Computer Simulation Paradigm. In *Proceedings of the 2012 ACM/IEEE/SCS 26th Workshop on Principles of Advanced and Distributed Simulation (PADS 2012).* ACM Press, NY, pp. 71-74.

Hong J and Luo J (2011). Large-Scale Ranking and Selection Using Cloud Computing. In *Proceedings of the 2011 Winter Simulation Conference.* ACM Press, NY, 4046-4056.

Rossetti M D and Chen Y (2012). A Cloud Computing Architecture for Supply Chain Network Simulation. In *Proceedings of the 2012 Winter Simulation Conference*. ACM Press, NY, 1-12.

Sakellari G and Loukas G (2013). A Survey of Mathematical Models, Simulation Approaches and Testbeds used for Research in Cloud Computing. *Simulation Modelling Practice and Theory* **39**:92-103.

Zhao W, Peng Y, Xie F and Dia Z (2012). Modeling and Simulation of Cloud Computing: A Review. In *Proceedings of the 2012 IEEE Asia Pacific Cloud Computing Congress (APCloudCC)*, pp. 20-24.

Kacsuk P, Farkas Z, Kozlovszky M, Hermann G, Balasko A, Karoczkai K, and Marton I (2012). WSPGRADE/gUSE generic DCI gateway framework for a large variety of user communities. *Journal of Grid Computing* **10(4)**: 601-630.

Velte T, Velte A and Elsenpeter R. *Cloud Computing, a Practical Approach.* McGraw-Hill, Inc, 2009.

Onggo B S S, Taylor, S J E and Tulegenov A (2014). Web-Based Simulation as a Collaborative Modelling and Community Development Tool. In *Proceedings of the 2014 Operational Research Society Simulation Workshop*, pp 103-112.

Grance T and Mell P (2011). The NIST Definition of Cloud Computing. NIST [Online]. Available at: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf [Accessed 23/12/12].

Taylor S J E, Fishwick P A, Fujimoto R, Page E H, Uhrmacher A M, Wainer G (2012). Panel on Grand Challenges for Modeling and Simulation. In *Proceedings of the 2012 Winter Simulation Conference*. ACM Press, NY., Article No. 232.

Taylor S J E, Khan A, Morse K L, Tolk A, Yilmaz L and Zander J (2013a). Grand Challenges on the Theory of Modeling and Simulation. In *Proceedings of the 2013 Symposium on the Theory of Modeling and Simulation*. SCS, Vista, CA., Article 34.

Taylor S J E, Balci O, Cai W, Loper M, Nicol D and Riley G (2013b). Grand Challenges in Modeling and Simulation: Expanding Our Horizons. In *Proceedings of the 2013 SIGSIM-PADS Conference*, ACM Press, NY, pp. 403-408.

Taylor S J E, Brailsford S, Chick S E, L'Ecuyer P, Macal C M and Nelson B (2013c). Modeling and Simulation Grand Challenges: An OR/MS Perspective. In *Proceedings of the 2013 Winter Simulation Conference*. ACM Press, NY. 2013c, pp. 1269-1282.

AUTHOR BIOGRAPHIES

SIMON J E TAYLOR is the convener of this new phase of Grand Challenge activities. He is the Founder and Chair of the COTS Simulation Package Interoperability Standards Group under SISO. He is the Editor-in-Chief of the UK Operational Research Society's (ORS) Journal of Simulation and the Simulation Workshop series. He was Chair of ACM's SIGSIM (2005-2008) and is a member of the SIGSIM Steering Committee. He is a Reader in the Department of Information Systems and Computing and Brunel and leads the ICT Innovation Group. He has published over 150 articles in modelling and simulation. His recent work has focused on the development of standards for distributed simulation and grid- and cloud-based simulation in industry (www.cloudsme.eu) as well as the spread of the international Grid Infrastructure into Africa (www.eI4Africa.eu). His email address is simon.taylor@brunel.ac.uk and his web address is www.brunel.ac.uk/~csstsjt.

TAMAS KISS is a Reader at the Department of Business Information Systems, Faculty of Science and Technology, University of Westminster. He holds a PhD in Distributed Computing and his research interests include distributed and parallel computing, cloud, cluster and grid computing. He has been involved in several European research projects such as CoreGrid, EDGeS, EDGI, SHIWA, and SCI-BUS, and FP7 support action projects such as DEGISCO, ER-flow and IDGF SP. He led and coordinated application support activities and work packages in these projects. Currently he is Project Director of the FP7 CloudSME project that develops a cloud-based simulation platform for manufacturing and engineering SMEs. He co-authored more than 80 scientific papers in journals, conference proceedings and as book chapters. His email address is T.Kiss@westminster.ac.uk.

PETER KACSUK is the Director of the Laboratory of the Parallel and Distributed Systems in the Computer and Automation Research Institute of the Hungarian Academy of Sciences. He received his MSc and university doctorate degrees from the Technical University of Budapest in 1976 and 1984, respectively. He received the kandidat degree (equivalent to PhD) from the Hungarian Academy in 1989. He habilitated at the University of Vienna in 1997. He recieved his professor title from the Hungarian President in 1999 and the Doctor of Academy degree (DSc) from the Hungarian Academy of Sciences in 2001. He served as full professor at the University of Miskolc and at the Eötvös Lóránd University of Science Budapest. He has been a part-time full professor at the Cavendish School of Computer Science of the University of Westminster. He has published two books, two lecture notes and more than 200 scientific papers on parallel computer architectures, parallel software engineering and Grid computing. He is co-editor-in-chief of the Journal of Grid Computing published by Springer.

GABOR TERSTYANSZKY is a Professor of Distributed Computing at the Faculty of Science and Technology. He is the Director of the Centre for Parallel Computing. Before joining Westminster in 2000. His research in the field of distributed and parallel computing addresses the interoperability of large-scale distributed computing infrastructures and workflows. Working closely with diverse scientific communities, he has contributed to breakthroughs in interoperability research that have enabled researchers in different fields to more effectively exploit computing infrastructures in their scientific experiments and to execute computationally-hungry and data-demanding applications. it takes to produce research results, and producing many more research results than before. Professor Terstyanszky has supervised more than ten PhD students. He attracted in excess of £1.4 million in European and UK research funding to the University of Westminster. He has led or contributed to numerous collaborative research projects funded by European Commission Research Framework Programmes and by UK Research Councils. He has published more than 100 papers in academic journals and in proceedings of European and international conferences. He is regularly invited to join program committees of scientific conferences, and reviews scientific papers in academic journals on a regular basis. His email address is G.Z.Terstyanszky@westminster.ac.uk.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

THE USE OF MASSIVELY PARALLEL PROCESSORS IN SIMULATION: AN ASSESSMENT

Dr. Russell C. H. Cheng

University of Southampton
Highfield, Southampton, SO17 1BJ
R.C.H.Cheng@soton.co.uk

ABSTRACT

In recent years there have been rapid advances in the design and availability of general purpose graphics processing units (GPGPUs) suitable for general purpose programming. Areas where they have been heavily applied include particle physics and financial modelling. However, so far, their use in discrete-event simulation modelling seems more limited. This paper assesses the usefulness of such hardware in OR simulation studies. Our initial conclusion is that their use in discrete simulation is potentially quite valuable, but in the more specific areas of input and output analysis in simulation experimentation their use can significantly increase the speed with which calculations can be done, especially when large data samples are involved. Handling "big data" samples has become increasingly important and the use of GPGPUs is likely to become ever more worthwhile. Examples are presented showing the improvement in computational speed possible.

Keywords: CUDA, GPGPU, maximum likelihood

1 INTRODUCTION

The use of parallel computing in discrete event simulation (DES) work has been well discussed in the literature. For a review see Fujimoto (1990). However it has only been in the last three years that has seen the availability of hardware based on graphics processing units (GPUs) that makes personal desktop parallel computing potentially widely available. A good general introduction to the field in general is given by Kirk and Hwu (2013).

A market leader in this field is CUDA (Computer Unified Device Architecture) a massively parallel computer platform and programming language, the latter being a relatively simple extension of C, which has become widely used in partial differential equation applications arising mainly in the physical sciences, see Rumpf and Strzodka (2006). It has also been used in financial modelling applications (see for example Pagès and Wilbertz 2012) which deploy related differential models. CUDA has also been used in statistical methodology, see Matloff (2013, Chap. 14) . To date its use in DES has been more limited.

In this paper we examine how massively parallel computation with CUDA might be used in simulation experimentation. Our main conclusion is that, though the hardware architecture would seem to handicap enhancing the performance of DES models themselves, in practice the gain in computational speed can be impressive. This is particularly so in the more specific context of simulation input or output analysis where it can significantly increase processing speed, especially when analysis of large data sets are involved. The aim of this paper is to examine how GPU hardware now available can be used in this way. A practical example is given involving the fitting of a finite mixture model with normally distributed components to a large real data set obtained from a manufacturing study showing speed increases well over 40-fold are possible using such a GPU. Simple DES models of the M/M/1 queue are also considered where speed increases of 30-fold are obtainable.

In the next section we describe the main features of the CUDA parallel computing platform, and in Section 3 we illustrate its use in a genuinely practical example involving fitting the finite mixture model, whilst in Section 4 a simple M/M/1 queue is considered.

2 MASSIVELY PARALLEL PROGRAMMING WITH CUDA

To make our discussion specific we consider only CUDA as it is set up on the particular massively parallel processing platform implemented in Tesla general purpose graphics processor units (GPGPUs) produced by the NVIDIA Corporation, one of the leading manufacturers in the field. Though Tesla GPUs have their origins in graphics cards, they are the first generation of NVIDIA cards designed with general purpose computing in mind. It has to be emphasized however that, though designed for general purpose programming, their architecture, called *Fermi*, is still strongly influenced by their graphics origin and this in turn determines to a large extent the form that the parallel programming code must take.

A very clear introduction to GPU programming with CUDA is provide by Matloff (2013, Chap. 5). We do not go into full details, but summarize some key points here.

A Tesla GPU is made up of an array of identical *streaming multiprocessors* (SMs). Each SM comprises a given number of individual *streaming processors* (SPs) usually called *cores* for short. In operation each core executes its own piece of code called a *thread*. Threads are arranged in *blocks* and the blocks are arranged in a *grid*.

The GPU is configured to operate with an accompanying CPU called the *host*, whilst the GPU is referred to as the *device*. The flow-controlling CUDA program is based in the host. The thread code is held in a routine called a *kernel* which is held in the device. The usual form that an application takes when running just one kernel is given in Figure 1 showing that the host simply has to place input data into device memory then *launch* the kernel, i.e. set the kernel running, then wait for the kernel to complete execution after which control is passed back to the host.

Host
H1 Calculate grid and block size to be used on the device.

H2 Allocate host memory for host input array.
H3 Allocate host memory for host output array
H4 Allocate device memory for device input array
H5 Allocate device memory for device output array

H6 Put input data into host input array
H7 Copy this data into device input array

H8 Launch kernel on the device using the calculated grid and block sizes.

Device
D1 Execute the threads in the device using device input array as input.
D2 Put output from threads into device output array
D3 Copy this to host output array
D4 Return control back to host

Figure 1 *Typical form of a one kernel application*

An application can run several kernels. If kernels are called one after another the arrangement is as in Figure 1, except that after step D2 control returns to the host which launches the next kernel. Threads of the new kernel can use the output from the previous kernel as its input. This avoids unnecessary copying of data arrays, a very important point in view of the way memory is configured and accessed by the GPU, which we consider next.

The device memory has a quite complicated hierarchical form with six levels. The two most important forms are *shared memory* and *global memory*. Each SM has its own shared memory, which as the name implies, is shared by all threads running in that SM. This memory is on-chip so access is fast, but the memory size is small being typically only 16k bytes. This memory cannot be accessed by the host. Global memory is off-chip and shared by all threads of an application and accessible from the host; the contents are held for the entire application and across more than one kernel. However access is very slow involving hundreds of clock cycles per access. It is therefore very important that an application does not require threads to be constantly accessing global memory, as this will usually result in loss of all the speed gained by parallel running.

The other key feature to note is that threads in a given SM *do not run independently*. Blocks are assigned to specific SMs, with, in most applications, more than one block being assigned to each SM. Moreover the threads in a block are grouped into sets of 32, each called a *warp*. The key point to note is that each SM always runs the threads of a given warp together, with the threads having to execute instructions in what is called *lockstep*. This means that all the threads have to carry out the same instruction at the same time. The only variation allowed is that a thread must skip an instruction it cannot perform. Execution of warps is time-sliced with this actively controlled by their SM; if the threads of the currently active warp cannot be run, for example because they are waiting to receive data from global memory, then the SM will switch to another warp making active one whose threads can be executed. Warp *latency* is reduced if the blocksize is sufficiently large so that warps can be queued for execution. However if efficiency is to be maintained it is far more preferable that threads of a warp all execute the same code, and that this code does not contain any branching. This avoids what is termed *divergence* from occurring with different threads having to wait for long periods to do quite different things.

This lock-step mode of operation would seem to prevent the possibility of different cores in the same SM from efficiently making its own run of a DES model. We have investigated this experimentally by making simultaneous but independent runs of a model of the M/M/1 queue on separate threads. We have tried with a model capable of running in lock-step mode, and another where the model does not run in lock-step. The results, which are discussed more fully in Section 4, appear promising with good speed up using either model.

We consider also an aspect of DES work where lockstep parallelization will be highly effective. This occurs in the statistical analysis of input or output data used or generated in a simulation experiment. Consider a sample of input or output data $\mathbf{y} = \{y_1, y_2, ..., y_n\}$. Statistical examination of such a sample frequently involves subjecting each observation y_j to precisely the same set of calculations. Lock-step parallel implementation of the calculations is then not only feasible, but is quite easy to implement.

We shall give a detailed example of such an application in the next section. However before we do so we summarize and illustrate our discussion of this section by describing the particular Tesla GPU used in the numerical examples to be presented below.

The examples make use of the C2075 Tesla GPU. This has 14 SMs each with 32 cores making 448 cores in all. Each core runs at a frequency of 1.15 GHz, delivering a peak double precision floating point performance of 515 Gflops and a single precision performance of 1.03 Teraflops. As a warp always has 32 threads, having 32 cores per SM makes choice of blocksize rather straightforward. If we choose blocksize to be a multiple of 32 then each block divides neatly into a whole number of warps. In our experimentation we used blocksizes ranging from 64 to 512, not finding the precise value to be all that critical for good performance. The GPU has 6 Gigabytes of GDDR5 memory.

At this time of writing the C2075 card is less than 3 years old, but is already considered by NVIDIA to be ‘legacy’ hardware. It is certainly not the most powerful GPU available. However it is designed specifically for general purpose computing rather than just graphical rendering, and is sufficiently powerful to give some idea of the kind of benefits possible using massively parallel processing both in DES modelling and in statistical input/output analysis, especially for large data sets. We consider a statistical application in the next section.

3 A STATISTICAL ESTIMATION EXAMPLE

3.1 Calculating a Loglikelihood

As an example of our discussion in the previous section we consider the calculation of an important quantity used in many problems of statistical inference, namely the log-likelihood function

$$L(\boldsymbol{\theta},\mathbf{y})=\sum_{j=1}^{n}\log(f(y_j,\boldsymbol{\theta})) \tag{1}$$

of a random sample $\mathbf{y}=\{y_1,y_2,...,y_n\}$ of observations drawn from a continuous probability distribution with probability density function (PDF) $f(y,\boldsymbol{\theta})$ assuming this depends on a vector $\boldsymbol{\theta}=(\theta_1,\theta_2,...,\theta_p)$ of parameters. A good way to estimate $\boldsymbol{\theta}$ when it is unknown is the well-known method of maximum likelihood (ML). The estimator is that value of $\boldsymbol{\theta}$ which maximizes the likelihood function, that is

$$\hat{\boldsymbol{\theta}}=\arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta},\mathbf{y})\,. \tag{2}$$

In simple cases $\hat{\boldsymbol{\theta}}$ may be given by an explicit formula. For example, for the normal model $N(\mu,\sigma^2)$, the maximum likelihood estimators are

$$\hat{\mu}=\bar{y}=n^{-1}\sum_{j=1}^{n}y_j,\ \hat{\sigma}^2=n^{-1}\sum_{j=1}^{n}(y_j-\bar{y})^2\,.$$

However for a sample drawn from the gamma distribution $G(\alpha,\beta)$ with PDF

$$f_G(y,\alpha,\beta)=\Gamma^{-1}(\alpha)\beta^{-\alpha}y^{\alpha-1}\exp(-y/\beta)$$

the MLE is not available explicitly and has to be found numerically. A very convenient general numerical optimization method for doing this is the well-known simplex search procedure proposed by Nelder and Mead (1965) calculates the loglikelihood for a sequence of parameter values $\boldsymbol{\theta}_0,\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,...,\boldsymbol{\theta}_m$, which converges to $\hat{\boldsymbol{\theta}}$, the process stopping once $\boldsymbol{\theta}_m$ is judged sufficiently close to $\hat{\boldsymbol{\theta}}$.

The amount of computational effort needed clearly depends on the sample size *n*, on the number of values of the loglikelihood that need calculating, and on how complicated the expression for the loglikelihood is. In the gamma case the loglikelihood can be written as

$$L_G(\alpha,\beta,\mathbf{y})=-n[\log(\Gamma(\alpha))-\alpha\log\beta]-(\alpha-1)\sum_{j=1}^{n}\log(y_j)-\beta^{-1}\sum_{j=1}^{n}y_j\,.$$

It will be seen in this case that the y_i appear only in the last two summations. These have only to be calculated once, at the start of the Nelder-Mead routine when the first loglikelihood value, $L_G(\alpha_0,\beta_0,\mathbf{y})$, is calculated at the first parameter setting, $\boldsymbol{\theta}_0=(\alpha_0,\beta_0)$, and do not need recalculation when calculating the loglikelihood at subsequent parameter values $\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,...,\boldsymbol{\theta}_m$. When *m* is large, the overall computational effort in obtaining $\hat{\boldsymbol{\theta}}$ is therefore not greatly affected by the size of *n*.

However for a sample from the Weibull distribution $W(\alpha,\beta)$ with PDF

$$f_W(y,\alpha,\beta) = \alpha\beta^{-\alpha} y^{\alpha-1} \exp[-(y/\beta)^{\alpha}],$$

the loglikelihood takes the form

$$L_W(\alpha,\beta,\mathbf{y}) = n[\log\alpha - \alpha\log\beta] + (\alpha-1)\sum_{j=1}^{n}\log(y_j) - \beta^{-\alpha}\sum_{j=1}^{n} y_j^{\alpha}.$$

In this case the last summation $\sum_{j=1}^{n} y_j^{\alpha}$ will have to be calculated separately at each $\boldsymbol{\theta}_m$ so that the Nelder-Mead process is now strongly linearly dependent on *n*. The problem is exacerbated if the number of parameters is large.

We consider in detail a situation discussed by Cheng and Currie (2003) where a finite mixture model with *k* normally distributed components is fitted. This has PDF

$$f(y,\boldsymbol{\theta}) = \sum_{i=1}^{k} w_i f_N(y \mid \mu_i, \sigma_i) \tag{3}$$

where $w_i > 0,\ i = 1, 2, ..., k$ with

$$\sum_{i=1}^{k} w_i = 1 \tag{4}$$

are the weights of the components whose individual densities are

$$f_N(y \mid \mu_i, \sigma_i) = \left(\frac{1}{2\pi\sigma_i^2}\right)^{1/2} \exp\left(-\frac{1}{2\sigma_i^2}(y-\mu_i)^2\right). \tag{5}$$

Here $\boldsymbol{\theta} = (\mu_1, \sigma_1, w_1, \mu_2, \sigma_2, w_2, ..., \mu_k, \sigma_k, w_k)$ so there are 3*k* parameters. In a similar way to the Weibull example, the individual terms $\log f(y_j, \boldsymbol{\theta})$ in the summation (1) have all to be recalculated for each $\boldsymbol{\theta}_m$, so there is a strong linear dependence on *n*.

3.2 Parallel Calculation of the Loglikelihood

We now consider how calculation of loglikeliood (1) can be done for the finite mixture model case case (3) using GPGPU methods. There are two main steps, both amenable to parallelization:

(i) Calculate the individual terms $l_j = \log f(y_j, \boldsymbol{\theta}),\ j = 1,2,..., n$ with $f(y_j, \boldsymbol{\theta})$ as defined in (3), (4), (5).

(ii) Calculate the sum $\sum_{j=1}^{n} l_j$.

The first step is very straightforward to do using parallel programming and can be carried out using just one kernel. This is because exactly the same calculations are used in evaluating each $l_j,\ j = 1,2,..., n$. The kernel therefore submits precisely *n* threads for processing by the device, with each thread calculating the value of precisely one l_j. As the form of the calculation is the same for every l_j this ensures that all active cores of a warp automatically work in lockstep. The thread code, given below (where $0.39894 \cong 1/\sqrt{2\pi}$), is quite simple:

```
__global__ void LogLikelihoodKernel(double* y, double* mu, double* sig, double*
w, double* Res, int n, int k)
{
   // Each thread computes one term of the loglikelihood
   // adding it to the Res vector
```

```
        int i;
        double temp = 0;

        int j = blockIdx.x * blockDim.x + threadIdx.x;
        if(j >= n) return;

        for(i=1; i<=k; i++)
        {
           temp += w[i] * (0.39894/sig[i]) * expf(-(y[j] - mu[i]) * (y[j] - mu[i])
                            /(2.0 * sig[i] *sig[i]) );
        }

        Res[j] = logf(temp);

        __syncthreads();
    }
```

The integer *j* is the subscript of the element in the block whose value is being calculated. In our implementation we used a block size of between 64 and 512 threads. The functions expf() and logf() are fast versions of the exponential and logarithmic functions implemented in CUDA. The final instruction, `syncthreads()`, makes the kernel wait until all threads are finished before continuing.

The second step involves two or more kernels to evaluate a sum using a well known construction in parallel processing known as *reduction*. The basic idea is most easily explained when $n = 2^d$. The reduction is carried out recursively using *n*/2 threads initially. In step #1 each thread adds 2 terms yielding *n*/2 subtotals. In step #2 just half the threads, that is *n*/4 threads, are used each to add 2 of the *n*/2 subtotals. The process continues until step #*d* when just one thread adds the final 2 subtotals to give the required overall total. The process has therefore taken $\log_2 n = d$ steps. The case where *n* = 8 is illustrated in Figure 2.

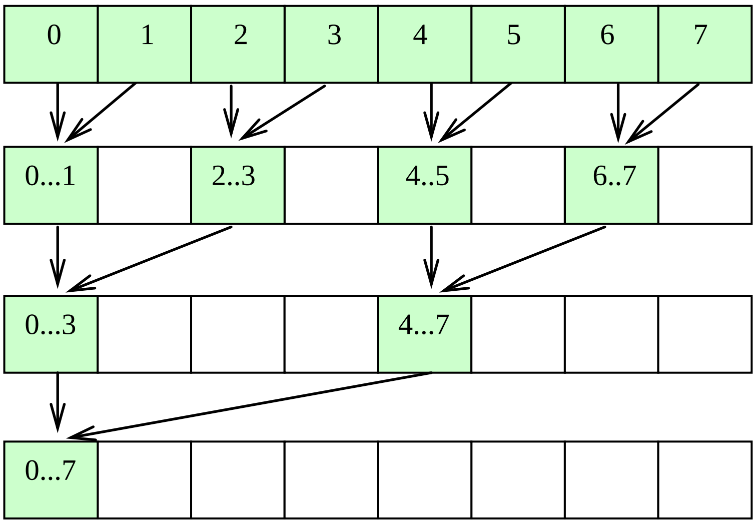


Figure 2 *Reduction process for adding 8 numbers in* $log_2 8 = 3$ *steps*

The process is somewhat more complicated when *n* is not an integer power of 2. See Lensch and Strzodka (2008) for details.

We have carried out a pilot experiment using a sample size of *n* = 32768 and using a ten-component model, that is with *k* = 10, to calculate the loglikelihood function of the finite mixture model with normal components as given by equations (3), (4) and (5). We used a ten-component model, i.e. *k* = 10, and timed how long it took to calculate the loglikelihood 1,000 times using first the host and then the device. The host CPU, an Intel G2030 processor running at 3.00 GHz, took 11.19 seconds, whilst the Tesla C2075 took 0.094 seconds using a gridsize of 64 and blocksize of 512. The device was therefore 119 times faster. Given that the cores have a clockspeed of 1.15 GHz, it would appear that the speed improvement is about 70% of the maximum achievable.

The likelihood is not usually calculated in isolation so the speed improvement just described represents an upper limit to what is achievable when calculation of loglikelihood values is only part of an overall statistical analysis carried out in a practical application. In the next section we discuss such an application, where calculations are essentially serial, but which require repeated calculation of the loglikelihood. Though there is a drop off in overall speed, repeated calculation of the loglikelihood forms the major part of the computational effort so that evaluating the loglikelihood using parallel routines is well worthwhile.

3.3 Results

A fully working application has been developed using an Excel front-end interface for fitting the finite mixture model. Though not discussed here this working version takes a Bayesian approach fitting the k component normal mixtures for a selected range of k: $k = 1,2,\ldots,k_{\max}$ where $k_{\max}$ is selectable. The Bayesian Information Criterion is also calculated for the maximized probability for each k. This allows the fits to be compared so that a best k can be selected.

We have used this application to fit the normal mixture model to an actual data sample comprising observations of 32151 activity cycle times obtained in a large study of a vehicle manufacturing process. There was some evidence that the activity cycle time contained different distributional components so that the frequency histogram of the distribution was not unimodal. A finite mixture model was therefore an appropriate distribution to consider fitting. In our experiment we not only fitted the model to the full sample where n = 32151, but also to subsamples of size n = 500, 1000, 2000, 4000, 8000 and 16000 randomly selected from the full sample. The application fitted the k-component mixture model for $k = 1,2,\ldots,k_{\max} = 8$. For each k the parameters were estimated numerically by ML estimation, using the host CPU, and using the Tesla C2075. Figure 3 shows the fitted distribution with k = 5 components for the subsample of size 2000.

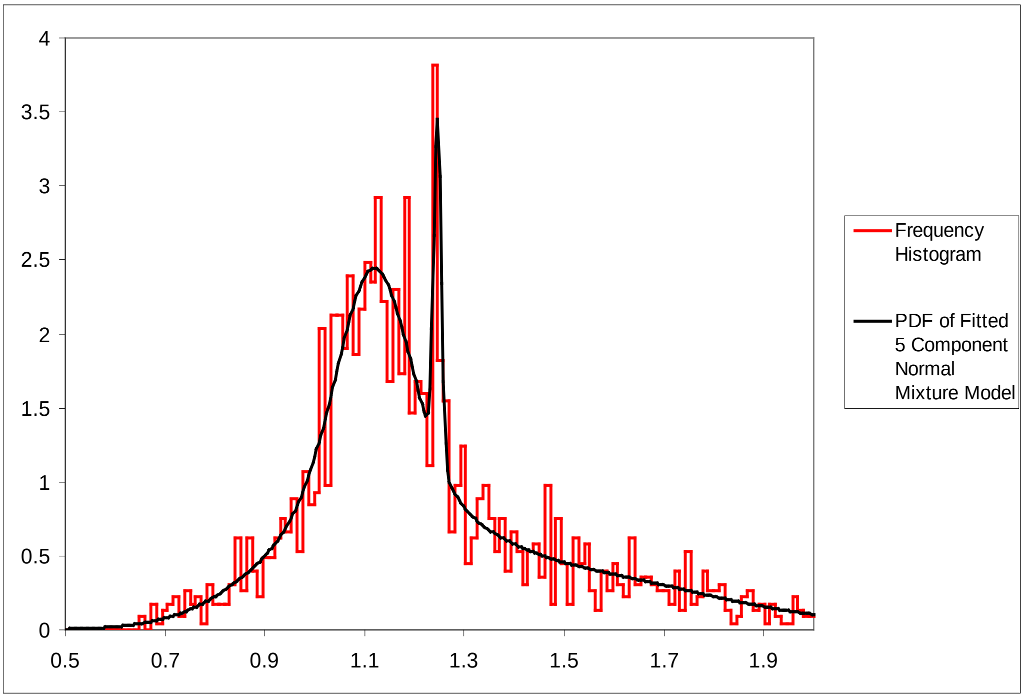


Figure 3 *PDF of 5-component normal mixture model fitted to a subsample of 2000 observations*

The times taken to carry out the estimation for all *k* values was recorded and these are shown in Table 1. It will be seen that using the GPU with parallel computation becomes progressively more effective. For the full sample size of *n* = 32151, the CPU takes over half-an-hour to fit the parameters of all the eight k-component models, whereas the GPU can do this in well under a minute.

Table 1 *Total Computation Time in Seconds for fitting eight k-component mixture models, k=1,2,...,8*

n	500	1000	2000	4000	8000	16000	32151
CPU	14.6	18.3	44.7	106.0	83.0	901.5	1906
GPU	6.03	5.88	7.0	8.3	5.3	22.5	40.3
CPU/GPU	**2.4**	**3.2**	**6.4**	**12.8**	**15.7**	**40.1**	**47.3**

4 M/M/1 MODEL

We now consider use of the Tesla 2075 GPU in DES. We discuss only the M/M/1 queue. However we consider two models. One is a model which can be run in lock-step, the other is a more standard DES model which cannot be run in lock-step.

The lock-step model, calculates the individual waiting times, that is the time from arrival until departure, of a prescribed number, *numcustomers*, of customers, assuming the queue is initially empty. We write W_i for the waiting time , S_i for the service time and T_i for the time between the arrival of the $(i-1)$ th and *i*th customers. Then the simple formulas

$$\begin{aligned} W_i &= (W_{i-1} + S_i - T_i) \quad \text{if } W_{i-1} \ge T_i \\ W_i &= S_i \quad\quad\quad\quad\quad\quad\;\; \text{if } W_{i-1} < T_i \end{aligned}$$

allow the W_i to be calculated in lock-step form using the following kernel function:

```
__global__ void gpu_mm1( float lam, float mu, float numcustomers, float numruns,
float *estimate, curandState *states)
{
  unsigned int tid = threadIdx.x + blockDim.x * blockIdx.x;
  if( tid < numruns )
  {
    long i; float s; float t; float u; float v;
    curand_init(1234, tid, 0, &states[tid]); // Initialize CURAND

    float w = 0;
    float totw = 0.;
    for(i=0; i< numcustomers; ++i)
    {
      u = curand_uniform (&states[tid]);
      s = -logf(u)/mu;
      u = curand_uniform (&states[tid]);
      t = -logf(u)/lam;
      v = w + s - t;
      w = w < t ? s : v; // this is the ternary op: if(w<t){w=s} else{w=v}
      totw += w;
    }
    estimate[tid] = totw/numcustomers;  //estimate of average waiting time
  }
```

```
}
```

The code is self explanatory, except to note that `curand_uniform()` is a CUDA library pseudo uniform random variable generator with a period of over 2^{190} values. It is initialized by the call to `curand_init` which takes *seed*, *sequence*, *offset*, and *address of a state* as arguments. The *state* is initialized by the initialization call. The initialization sets the state to the state that would be arrived at after (2^{67}.*sequence* + *offset*) calls to `curand_uniform()` starting from the seed state. Thus use of the thread index *tid* for the *sequence* value means that each thread will sample up to 2^{67} ($>1.475\times10^{20}$) values in its own segment of the total period before repeating values from another segment. Launching this kernel with gridsize = 14, blocksize = 512, numcustomers = 50000 produces 14*512 = 7168 runs of the queue completed in 0.811 seconds compared with 24.27 seconds for making the same number of runs of the same length carried out on the host CPU; a near 30-fold increase in speed when using the GPU.

We also ran a more standard version of the model. We do not give the complete kernel but just the main simulation loop included in the kernel to indicate its “if...else...” structure showing that, run in parallel, the code, given the inclusion of random sampling, will not be executable in lock-step.

```
    // Main simulation loop in the kernel
    while (0 == 0) {
      if (timenextarr < timenextdep){      // *** Event arrival ***
        time = timenextarr;
        s = s + n * (time - tn);  // Update area under "s" curve
        n++;
        tn = time;       // tn = "last event time" when next event is reached
        a++;
        if(a==numcustomers){     // do not schedule any more arrivals
          timenextarr = infinity;
        }
        else{
          u = curand_uniform (&states[tid]);
          interarrtime= -(1./lam)*log(u);
          timenextarr = time +interarrtime;
        }

        if (n == 1){
          u = curand_uniform (&states[tid]);
          servicetime= -(1./mu)*log(u);
          timenextdep = time +servicetime;
        }
      }
      else{                         // *** Event departure ***
        time = timenextdep;
        s = s + n * (time - tn);  // Update area under "s" curve
        n--;
        tn = time;          // tn = "last event time" when next event is reached
        c++;                // Increment number of completions
        if(c==numcustomers) goto endsimlabel;
        if (n > 0){
          u = curand_uniform (&states[tid]);
          servicetime= -(1./mu)*log(u);
          timenextdep = time +servicetime;
        }
        else { // noone in queue, so no departures to schedule
          timenextdep = infinity;
        }
      }
    } // end of main sim loop

endsimlabel:
    estimate[tid] = s/time;  //estimate of average queue length from this run
```

With the same settings as before, namely gridsize = 14, blocksize = 512, numcustomers = 50000, the kernel of 7168 threads executed in 1.2 seconds, whilst the host CPU version took 37.1seconds to run. Thus the GPU version again gave a speed up of over 30 times. Our conclusion is that running the code without lock-step did not result in any speed degradation in this simple DES example.

5 SUMMARY

The examples gives a good indication of the usefulness of desktop massively parallel computing in the statistical analysis of simulation experiments and in DES. It shows that when sample sizes are relatively large then an order of magnitude or more of improvement is achievable in computing speed.

Our results are sufficiently encouraging for us to recommend further more extensive investigation.

We have only discussed the CUDA programming platform. CUDA programs require detailed knowledge of the architecture of GPUs. The resulting computer code is heavily dependent on this architecture. Alternative approaches have been considered, and these are well discussed by Kirk and Hwu (2013). One, called *Thrust*, is already part of the CUDA software development kit and another is called *Arrayfire*. Both provide, mainly vector, routines for carrying out commonly occurring basic calculations, like the reduction process discussed in this paper. Parallelization is built into the Thrust and Arrayfire routines and is effectively hidden from the user, saving the user from having to grapple with the technical details of parallelization. Another approach is *OpenMP* and *OpenACC* which allow the user simply to insert directives or pragmas into their own non-parallel C-code to indicate to an appropriate compiler which segments of the code might benefit from parallelization. With this approach, the onus is on the compiler to introduce parallelization as it sees fit. All these approaches are attractive in that they free the user from having to carry out any parallel programming. However this does mean that the user does not know what form the parallelization takes so that the user cannot readily evaluate whether worthwhile improvement in speed will be achieved.

REFERENCES

Cheng R C H and Currie C S M (2003). Prior and candidate models in the Bayesian analysis of finite mixtures. In Chick S, Sanchez P J, Ferrin D and Morrice D J (eds). *Proceedings of the 2003 Winter Simulation Conference.* IEEE, Piscataway, pp 392-398.

Fujimoto R M (1990). Parallel discrete event simulation. *Communications of the ACM* **33(10):** 30-53.

Kirk D B and Hwu W-M W (2013). *Programming Massively Parallel Processors: A Hands-on Approach.* Elsevier: Waltham.

Lensch H and Strzodka R (2008). *Massively parallel computing with CUDA: Lecture #6.* http://www.mpi-inf.mpg.de/~strzodka/lectures/ParCo08/ accessed 2 February 2014.

Matloff N (2013). *Programming on parallel machines.* http://heather.cs.ucdavis.edu/~matloff/158/PL-N/ParProcBook.pdf accessed 2 February 2014.

Nelder J and Mead R (1965). A simplex method for function minimization. *Computer Journal* **7**, 308-313.

Pagès G and Wilbertz B (2012). GPGPUs in computational finance: massive parallel computing for American options. *Concurrency and Computation:Practice and Experience* **24(8):** 837-848.

Rumpf M and Strzodka R (2006). Graphics processor units: new prospects for parallel computing. In*:* Bruaset A M and Tveito A (eds). *Numerical Solution of Partial Differential Equations*. *Chapter 3, Lecture Notes in Computational Science and Engineering, v. 51*. Springer: Berlin Heidelberg pp 89-132.

AUTHOR BIOGRAPHY

RUSSELL C. H. CHENG is Emeritus Professor of Operational Research at the University of Southampton. He has an M.A. and the Diploma in Mathematical Statistics from Cambridge University, England. He obtained his Ph.D. from Bath University. He is a former Chairman of the

U.K. Simulation Society, a Fellow of the Royal Statistical Society and Fellow of the Institute of Mathematics and Its Applications. His research interests include: design and analysis of simulation experiments and parametric estimation methods. He was a Joint Editor of the *IMA Journal of Management Mathematics*. His email and web addresses are <R.C.H.Cheng@soton.ac.uk> and www.personal.soton.ac.uk/rchc.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

GENERATING INSIGHTS: THE EFFECTIVENESS OF DICRETE-EVENT SIMULATION MODELS IN CREATIVE PROBLEM SOLVING

Anastasia Gogi

School of Business and Economics
Loughborough University
Loughborough, LE11 3TU
a.gogi@lboro.ac.uk

Dr Antuela A Tako

School of Business and Economics
Loughborough University
Loughborough, LE11 3TU
a.takou@lboro.ac.uk

Prof Stewart Robinson

School of Business and Economics
Loughborough University
Loughborough, LE11 3TU, UK
s.l.robinson@lboro.ac.uk

ABSTRACT

It is often said that a discrete-event simulation model supports people in understanding and problem-solving by creating new and effective solutions, but little empirical evidence exists to substantiate this claim. While the discrete event community has not well-developed this philosophy of practice, empirical evidence from the system dynamics field shows that learning and generating new ideas from the use of models can be challenging. On the other hand, the alternative uses of visual features of simulation models seem not to have been exploited completely. Consequently, the role of discrete-event simulation models in creative problem solving is difficult to validate. This paper introduces the process of insight from the field of creative cognitive psychology and discusses its relation to creative problem-solving with discrete-event simulation. We conclude by discussing the methodology that will be employed to provide empirical evidence to support the value of discrete-event simulation models in creative problem-solving.

Keywords: Insight, Aha! Moment, Learning, Simulation

1 INTRODUCTION

Traditionally, the Operational Research (OR) community have built models for the purpose of predicting, and being prepared for, the future. In this analytical paradigm, models are evaluative in nature, while the decision-makers' role is passive. In particular, models are conceived as being predictive in nature and it is assumed that they are able to represent reality accurately. Consequently, the role of models is to provide guidance on how to be prepared for the future by systematically comparing possible courses of action.

However, this approach is not always applicable. Ackoff (1979) highlighted the need to shift our 'hard' OR view on the role of models and proposed to use them for the purpose of designing and inventing a better future. In this paradigm, models are conceived as means that aid people in understanding how a system currently operates and in creating new ideas on how this system can be 're-designed' in order to decide how to make that system operate better in future. Anecdotal evidence shows that the creation of new ideas may be based on intuition rather than relying on analysing systematically alternatives (Robinson, 2001). This process of generating creative ideas, especially by intuition, which addresses problems effectively, is called as the process of insight in the creative cognitive psychology literature (Mayer, 2010). Hence, in contrast to the traditional paradigm, in the design-and-invent paradigm models are insightful in nature and decision-makers are active.

Reviewing the last two decades' technological and methodological progress in computer simulation, it seems that the simulation community has taken into consideration Ackoff's suggestions. In particular, Visual Interactive Simulation (VIS) and Visual Interactive Modelling (VIM) are features that have been added in almost all simulation software packages, bringing decision-makers into the heart of the simulation project life-cycle (Bell and O'Keefe, 1987; Bell, 1991). The general belief, which led to these technical improvements in simulation packages, is that the more active the decision makers' role in a simulation project becomes, the better they will understand the problem situation (Kotiadis et al 2013; Rouwette et al 2011); and hence, the more creative they will be about the required changes to improve this situation (Robinson, 2004). In addition, scholars have developed frameworks to guide the process of building successful models identifying the socio-political factors that may influence the process and the quality of a simulation project (Robinson and Pidd, 1998; Harper and Pitt, 2004; Baldwin et al 2004). As such, it seems to be agreed that simulation interventions lead to creative problem solving, that is, helping people to create new and effective ideas.

Is there any evidence to support the usefulness of discrete-event simulation (DES) models in creative problem solving? To what extent do the animation and/or statistical features of Visual Interaction Modelling and Simulation (VIM&S) support people in creating new and effective solutions? The purpose of this paper is to explore these questions with reference to the concept of 'insight' as it is understood in cognitive psychology. First we demonstrate that little attention has been given to answering these questions. We then introduce the process of insight from the field of creative cognitive psychology and discuss its relation to creative problem-solving with DES. We conclude by discussing the methodology that will be employed to provide empirical evidence to support the value of DES models in creative problem-solving.

2 CAN THE USE OF SIMULATION MODELS FACILITATE THE GENERATION OF INSIGHTS?

Bayer et al (2010) recognise that a distinct role of a simulation model is to support people in problem understanding and problem solving, especially by exploring new ways to address problem situations (i.e. creative problem-solving). However, DES is generally viewed as a 'hard' OR technique. Although Robinson (2005) warns that alternative uses of DES are crucial for the future of the technique, van der Zee and Slomp (2009) are surprised by seeing: "how the dominant view on simulation use, as described in many course books and implemented in software, has hardly altered since then. It still considers the use of simulation as just a methodology to analyse design decisions". On the other hand, practitioners and scholars often discuss that DES interventions do support stakeholders to gain insights about their problems and subsequently to generate effective ideas on how to address them. However, these learning outcomes are often omitted from written case studies. Paul and Kuljis (2010) arrive to the same conclusion by claiming that simulation community places more emphasis on model solving rather than problem solving. As a result, it is hard to find cases in the literature in which DES models were 'generators' of new and effective ideas.

A number of works that are examples of alternative uses of DES are Robinson (2001), Dobson and Shumsky (2006), Monks et al (2014), van der Zee and Slomp (2009), Smith (1990) and Elder (1992). Although these authors illustrate that DES can be used for 'softer' applications, their focus is mainly on the first aspect of the process of creative problem solving i.e. problem understanding. Robinson (2001) does demonstrate that a softer approach can lead to both a better understanding of the problem situation and the creation of new and effective ideas during the intervention. However, the details of the role of simulation in generating new and effective ideas is not made transparent. Although the author makes clear that the simulation model was the catalyst for problem understanding, it is not clear whether the ideas that were generated were the product of the human-to-human and/or the human-to-model interaction. As a result, little empirical evidence exists crediting the role of a DES model in creative problem-solving.

Meanwhile, in the system dynamics field the use of simulation models is recognised as being much more intertwined with learning. Sterman (1994) believes that (system dynamics) simulation is the main, and perhaps the only, way to learn about how and why a system behaves in a particular way. Morecroft (1992) agrees and explains the reason he believes this is feasible is because "simulations amplify and clarify scenario thinking". In other words, people are expected to learn about the structure of a system and the effectiveness of possible strategies through experimenting with a simulation model.

Nevertheless, learning from simulation models appears more challenging than expected. Recent experimental studies show that simulation users appear unable to explain or intuitively predict the behaviour even of the simplest system. Scientists explain that people face this difficulty because they tend to misperceive dynamic rules and simplify the way a complex system is structured (Cronin et al 2009; Diehl and Sterman, 1995; Langley and Morecroft, 2004; Senge and Sterman, 1992; Sterman, 1989a; Sterman, 1989b; Warren and Langley, 1999). These empirical results are in alignment with behavioural decision theory which reveals that people suffer from bounded rationality because of a number of judgemental biases and heuristics that people employ in complex situations (Kahneman, 2011; Simon, 1979). As a result, users face difficulties in suggesting a strategy that actually improves a system's behaviour since they do not consciously understand the underlying principles of the system (Monxes, 2004).

On the other hand, even if simulation users consciously understand how and why a system behaves in a particular way, this knowledge does not necessarily lead them to the creation of new strategies which improve the behaviour of that system (Bakken et al 1992; Cavaleri and Sterman, 1997; Größler, 2001). To overcome these difficulties researchers suggest that the decision-makers are highly involved in the modelling process (Rouwette et al 2011) or that detailed instruction is given prior to the experimentation phase (Kopainsky et al 2009). However, any improvements in the performance of the real system cannot be credited to model use on its own. Consequently, the effectiveness of the use of simulation model as a means of creating new ideas cannot be validated.

Another reason that may have led to the lack of appraisal of simulation models as a means of creative problem-solving might be the under-exploitation of the features of VIM&S. In particular, the visual features of simulation models are mainly used for validation and demonstration purposes. Van der Zee and Slomp (2009), for instance, highlight that visualisation is predominantly used for verifying a model's correctness, validating and demonstrating simulation outcomes. Baldwin et al (2004) argue that although simulation offers good features to cope with problem understanding and solving, in practice these features have not been exploited as much as they should. Therefore, Baldwin et al suggest an alternative modelling approach which involves using the animated model at the initial stages of the simulation process. Kopainsky et al (2009) perform experimental research in which dynamic graphs are used as an alternative method to visually and persuasively demonstrate the structural relations in a model. Again, the emphasis is placed on using simulation models to better understand and communicate the primary issues of a problem and not to support the creation of new and effective ideas. Overall, the role of simulation models in creative problem solving, though assumed plausible, still remains an open question.

3 THE PROCESS OF GENERATION OF INSIGHT

From the analysis of the previous section it is apparent that the effectiveness of DES models in creative problem-solving needs to be further developed and studied. It would be useful to approach this issue from a different perspective, but centred around the idea of creativity. The approach must recognise the need for effectively addressing a problem as well as problem understanding. It should also be possible to operationalize the approach. A literature review undertaken in fields of Gestalt, creative cognitive psychology and a loose collection of studies on insight led us to the process of generating 'insight' as an effective means to assess this distinct role of simulation. We now describe this concept and consider how it might be brought into the simulation context.

3.1 What Is Insight?

The word 'Insight' is used in two ways. It is used as a state of understanding – that is, to gain insight into something (Dominowski and Dallob, 1995; Smith, 1995). In this respect, Webster's New World Dictionary (1989) defines insight as 'the ability to see and understand the inner nature of things clearly, especially by intuition'. Insight is also described as an experience, an Aha! experience, involving a moment of epiphany (Schooler et al 1995). This view is originally encountered in the story of Archimedes of Syracuse when he discovered the principle of displacement – 'eureka'. For this research we adopt this latter concept, proposing it as an alternative approach to measure the value of simulation as a means for creative problem solving.

To explore Aha! insight in more depth, relevant literature was reviewed. In particular, the theoretical domain of Gestalt theory (Maier, 1940; Dominowski and Dallob, 1995; Mayer, 1995), creative cognitive psychology (Sternberg, 2009; Chronicle et al 2004; Öllinger et al 2008; Seifert et al 1995; Ward et al 1999) and a loose collection of studies on insight that have attempted to conceptualise the phenomenon (Metcalfe and Wiebe, 1987; Schooler et al 1995; Topolinski and Reber, 2010; Bowden and Jung-Beeman, 2003; Bowden et al 2005; Kounios et al 2006; Kounios et al 2008). Despite the fact that these streams of literature do not share the same theoretical foundation, it seems that they all agree upon the phenomenological perspective of the concept. In summary, the process of insight shares some features with intuition. More explicitly, a widely accepted definition of intuition is "affectively charged judgements that arise through rapid, non-conscious and holistic associations" (Dane and Pratt, 2007). While the suddenness and rapid knowing, the involuntary and spontaneous moment, and the intense emotional feeling are all attributes of intuition, they are also encountered in the related, yet distinct, construct of insight (Akinci and Sadler-Smith, 2012; Gunnells, 2011). Insight, in the form of an Aha! experience, is a sudden and affectively informed moment of creativity (Sternberg, 1999). However, the complexity of the process as well as the consciousness in the process are features that distinguish insight from intuition. In particular, generating insights, in contrast to the process of intuiting, is a long and slow process as it involves an impasse and an incubation phase (Dane and Pratt, 2007). An impasse refers to the conscious realisation that reproductive thinking cannot address this problem (Bowden et al 2005; Dominowski and Dallob, 1995; Jones, 2003). Consequently, a pursuit of novel ways to address the problem follows. At this stage, an incubation phase is usually involved, which is a period where the mind may not be occupied with the problem consciously. If a problem solver manages to find a novel way to address the problem during this incubation phase, then an involuntary moment of illumination has occurred (Hélie and Sun, 2010; Mayer, 1995; Smith, 1995). As insight is, therefore, the outcome of incubation, it might be expected that the suggested solution is based on 'gut feeling'(i.e. this is the case with intuition). Nevertheless, when a problem solver generates insights, s/he has a clear and conscious understanding about why s/he could not solve initially the problem and can explicitly explain how the problem can be solved. (Mayer, 2010; Sternberg, 1999).

In terms of the cognitive mechanisms that underlie insightful problem solving, scientists have offered many explanations which seem somewhat interrelated. More specifically, it is believed that prior knowledge and experience constrain people's worldview, and, as a result, this knowledge may prevent them from seeing the world as it really is. To overcome this mental fixation, insight may occur when people change the representation of a problem or restructure problem elements (Durso et al 1994; Davidson, 1995; Kershaw and Ohlsson, 2004; Öllinger et al 2013; Weisberg, 1995). Such a change in representation may also involve a reinterpretation of some problem elements or may form new goals and decision rules. However, in some cases past experience may not always inhibit creative thinking. Instead, prior knowledge may be used as a building block with which new ideas can be created (Weisberg, 1999). This is particular relevant when problem solvers have an Aha! experience through finding an analogy to a novel problem (Davidson, 1995; Finke, 1996; Mayer, 1995).

Furthermore, memory seems to be closely interrelated with the emergence of insight. Some theories argue that an insight emerges through forgetting the elements that fix one's perception about the problem (Davidson, 1995; Smith, 1995). On the other hand, other empirical research shows that insights may occur through remembering prior problem solving attempts as in this way problem solvers are led to the impasse more quickly (Kaplan and Simon, 1990). The environment that surrounds the problem-solver, such as colleagues' relevant work or inspirational clues, is also considered as a factor that influences the occurrence of insight (Csikszentmihalyi and Sawyer, 1995; Dunbar, 1995). Other factors such as personality attributes, intelligence and intrinsic motivation are considered to be influential upon creative behaviour (Kounios and Beeman, 2009), but empirical research does not show that they have a prime role in the process of insight (Davidson, 1995; Gunnells, 2011).

Unifying this literature, the process of insight generation may be defined as follows: insight, in the form of a Aha! experience, is "the process whereby a problem solver suddenly moves from a state of not knowing how to solve a problem to a state of knowing how to solve it" (Mayer, 2010; Schooler et al 1995). A satisfactory solution to the problem suddenly emerges after initial attempts to solve it have failed (i.e. impasse). This productive activity is about creating something new or novel. It uses past experience in a general way, but avoids being confined by habits or irrelevant associations. This affectively informed act of creativity is typically characterised by an Aha! experience. In achieving illumination, the problem solver variously overcomes mental constraints, changes mental representations, becomes aware of a new association between parts of a system, changes the meaning of some problem element or assimilates possible solutions from the environment.

3.2 Insight in the Simulation Context

Applying the above to the context of a DES project suggests that insight occurs when a simulation user suddenly knows how to improve the performance of a system after several failed alternative attempts to create an improvement with a simulation model (i.e. what-ifs scenarios). The strategy with which major improvements in the system are achieved then involves doing something new or novel. The model that attempts to describe the real life system as accurately as possible for the purpose of a study's goals is used as a basis for generating insight, although it may not be predictive in the sense of the traditional OR approach.

Demystifying the process of generation of insight into its components allows us to operationalise the phenomenon. For instance, we can identify that insight occurs during the experimentation phase of a simulation project if the users' problem solving pattern consists of the following:

- an impasse - failed initial attempts to improve model performance
- a sudden generation of new or novel ideas on how a model can be improved
- the confirmation that these ideas lead to major improvements to a problem situation, if not to its solution

Note that since a simulation model can only be a simplification of the real world, insights first concern the model and second, in a broader sense, the real-life system.

Bringing the possible causes of experiencing insights from the creative cognitive psychology literature to the context of DES, the following provides a non-exhaustive list of possible explanations for the occurrence of insight with a simulation model. From a cognitive perspective, an epiphany (Aha! moment) may emerge after a simulation user:

- overcomes implicitly posed constraints – for instance, in Robinson (2001) the stakeholders realise that there was no actual need for hiring lower skilled staff to work on the helpline.
- changes mental representation or/and identifies new relationships of variables – in Lee (2010), for example, students improved their thinking on how the supply of water can be increased when relationships among the important concepts of the water cycle were identified.
- changes the meaning of some problem element – for example, in Monks et al (2014) some participants succeed in ceasing a common misconception about the relationship between resource utilisation and service level.
- assimilates possible solutions from the environment – in Bakken et al (1992), for instance, some participants manage to apply insights learned from the first game to the second one.

4 PLANNED EMPIRICAL RESEARCH

The aim of our research is to determine whether the process of insight occurs with the use of DES models. In particular, we are going to focus on how the animation and statistical features of VIM&S

can support the emergence of insights. As such, the aim is to contribute towards better understanding how DES models support understanding and decision-making.

The next stage of our work involves an empirical study of generating insight from simulation models. In more detail, a controlled laboratory experiment will be conducted. This approach is beneficial because it allows the isolation and the determination of the features of VIM&S that promote insights. Also, because the task experiment can be replicated, it makes it possible to generate a larger sample size than from observing real life studies (Sternberg, 2009). In order to design and run successfully a laboratory experiment, however, careful thought and consideration must be given to the task, the participants, the independent and dependent variables (Größler, 2004).

It is essential to ensure that the task of the laboratory experiment resembles the insight problems that are used in the experiments conducted by creative cognitive psychologists. In particular, the task should be problem-solving oriented, whereas the solution to the task should require 'out of the box' thinking. This means that there will be a task goal which will be made known to participants before starting the experiment. Participants will then seek to find out how this goal can be achieved instead of merely appreciating the causes that lead the system to a certain behaviour. The pursuit of the solution must also be difficult, requiring the participant to approach the problem in a novel way. Furthermore, the task must resemble other familiar problems, be simple enough, and be relevant to participants' interests and knowledge (Wynder, 2004). It is important to provide monetary rewards which are not linked to task's goal attainment (Bonner and Sprinkle, 2002). These conditions are vital so as to maximise the possibility of observing a problem-solving procedure which involves an impasse, an incubation and an illumination. The monetary incentives also help to ensure a representative and satisfactory sample size (Abeler and Nosenzo, 2013; Bonner and Sprinkle, 2002). For this experiment, we are planning to base the insight problem around a model of the NHS111 service; a telephone service for non-emergency health care.

Students with operations management background are considered as appropriate participants for this experimental task. This decision is based on previous experimental research where both managers and students had to use simulation models to decide upon the strategy that could give the best results of a business problem (Bakken et al 1992). While it may be expected that managers would have outperformed students, the findings revealed the opposite. This is mainly because the managers' prior knowledge and experience constrained their free experimentation and learning from the simulation models. Consequently, managers were not willing to consider any new, risky, strategies. On the other hand, students, who had no prior experience and knowledge, were not afraid to take risks. As a result, they outperformed the managers. Our preference for students who have studied operations management is so that they have some understanding of the theory and practice of managing operations such as NHS111.

One independent variable will be manipulated: the features of the simulation model. Participants will be volunteers and randomly assigned in four different conditions to solve the same problem: without the simulation model, with statistical results generated from simulation runs, with an animation of the simulation, with both the animation and the statistical results. In terms of the dependent variable, the occurrence of insight is suggested to be measured with reference to the observation of the following problem-solving pattern: sudden shift from familiar yet failed attempts to solve the task to a distinct new attempt which leads directly to problem's goal attainment. For each condition, the subjects' problem-solving attempts will be recorded and analysed. If an insightful pattern is observed, then it means that the subject managed to solve the task with insight. If subjects solve the problem, but they reach the solution following a different pattern (e.g. a new approach was found but not in the right form at once), then the subjects' experience of insight is ambiguous. For that reason, pre- and post-experiment questionnaires will also be used to determine whether subjects arrive at the task's solution consciously and can provide satisfactory explanations about the reasons for the problem situation. Finally, if subjects do not manage to solve the problem, then insight must not have occurred as insight is directly linked with a problem's solution.

Our expectation is that participants who have access to both the animation and the statistical results will be more likely to follow an insightful pattern. Meanwhile, those who do not have access to the simulation at all are much less likely to experience insight.

5 CONCLUSION

It is assumed that the use of a DES model supports problem understanding and problem solving by creating new and effective ideas that address a problem. To understand the role of simulation as a means of creative problem solving, therefore, the process of generation of insight has been introduced. This approach is considered to be appropriate as conscious understanding, creativity and problem solving are all features of the process of insight. The next phase of our research will investigate from an empirical perspective the process of generating insight using simulation.

ACKNOWLEDGEMENT

We acknowledge the financial support and advisory input of SIMUL8 Corporation for this work.

REFERENCES

Abeler J and Nosenzo D (2013). Self-selection into economics experiments is driven by monetary rewards. *Institute for the Study of Labor (IZA)* Discussion Papers 7374, URL: http://ftp.iza.org/dp7374.pdf.

Ackoff R L (1979). The future of operational research is past. *Journal of the Operational Research Society* **30(2)**: 93-104.

Akinci C and Sadler-Smith E (2012). Intuition in Management Research: A Historical Review. *International Journal of Management Reviews*, **14(1)**: 104-122.

Bakken B, Gould J and Kim D (1992). Experimentation in learning organizations: A management flight simulator approach. *European Journal of Operational Research* **59 (1)**: 167-182.

Baldwin L P, Eldabi T and Paul R J (2004). Simulation in healthcare management: a soft approach (MAPIU). *Simulation Modelling Practice and Theory* **12(7–8)**: 541-557.

Bayer S, Bolt T, Kapsali M and Brailsford S (2010). The social role of simulation models. In: Moon T-H (ed). *Proceedings of the 28th International Conference of the System Dynamics Society*. System Dynamics Society: Seoul, Korea.

Bell P C (1991). Visual interactive modelling: the past, the present, and the prospects. *European Journal of Operational Research* **54(3)**: 274-286.

Bell P C and O'Keefe R M (1987). Visual interactive simulation—history, recent developments, and major issues. *Simulation* **49(3)**: 109-116.

Bonner S E and Sprinkle G B (2002). The effects of monetary incentives on effort and task performance: theories, evidence, and a framework for research. *Accounting, Organizations and Society* **27(4)**: 303-345.

Bowden E M and Jung-Beeman M (2003). Aha! Insight experience correlates with solution activation in the right hemisphere. *Psychonomic Bulletin & Review* **10(3)**: 730-737.

Bowden E M, Jung-Beeman M, Fleck J and Kounios J (2005). New approaches to demystifying insight. *Trends in Cognitive Sciences* **9(7)**: 322-328.

Cavaleri S and Sterman J D (1997). Towards evaluation of systems thinking interventions: a case study. *System Dynamics Review* **13(2)**: 171-186.

Chronicle E P, MacGregor J N and Omerod T C (2004). What makes an insight problem? The roles of heuristics, goal conception, and solution recoding in knowledge-lean problems. *Journal of Experimental Psychology. Learning, Memory & Cognition* **30(1)**: 14-27.

Cronin M A, Gonzalez C and Sterman J D (2009). Why don't well-educated adults understand accumulation? A challenge to researchers, educators, and citizens. *Organizational Behavior and Human Decision Processes* **108(1)**: 116-130.

Csikszentmihalyi M and Sawyer K (1995). Creative insight: the social dimension of a solitary moment. In: Sternberg R J and Davidson J E (eds). *The Nature of Insight*. The MIT Press: Cambridge, MA, US, pp 329-363.

Dane E and Pratt M (2007). Exploring intuition and its role in managerial decision making. *Academy of Management Review* **32(1)**: 33-54.

Davidson J E (1995). The suddenness of insight. In: Sternberg R J and Davidson J E (eds). *The Nature of Insight*. The MIT Press: Cambridge, MA, US, pp 125-155.

Diehl E and Sterman J D (1995). Effects of feedback complexity on dynamic decision making. *Organizational Behavior and Human Decision Processes* **62(2)**: 198-215.

Dobson G and Shumsky R (2006). Web-based simulations for teaching queuing, Little's Law, and inventory management. *INFORMS Transactions on Education* **7(1)**: 106-123.

Dominowski R L and Dallob P (1995). Insight and problem solving. In: Sternberg R J and Davidson J E (eds). *The Nature of Insight*. The MIT Press: Cambridge, MA, US, pp 33-62.

Dunbar K (1995). How scientists really reason: scientific reasoning in real-world laboratories. In: Sternberg R J and Davidson J E (eds). *The Nature of Insight*. The MIT Press: Cambridge, MA, US, pp 365-396.

Durso F T, Rea C B and Dayton T (1994). Graph-theoretic confirmation of restructuring during insight. *Psychological Science* **5(2)**: 94-98.

Elder M D (1992). *Visual interactive modelling: some guidelines for its implementation and some aspects of its potential impact on operational research*. PhD Thesis, University of Strathclyde.

Finke R A (1996). Imagery, Creativity, and Emergent Structure. *Consciousness and Cognition* **5(3)**: 381-393.

Gates E (1989). Webster's New World Dictionary. Cambridge University Press.

Größler A (2001). Musings about the effectiveness and evaluation of business simulators. In: Hines J H, Diker V G, Langer R S and Rowe J I (eds). *Proceedings of the 19th International Conference of the System Dynamics Society*. System Dynamics Society: Atlanta, USA.

Größler A (2004). Don't let history repeat itself? Methodological issues concerning the use of simulators in teaching and experimentation. *System Dynamics Review* **20(3)**: 263-274.

Gunnells K (2011). *Rational, intuition and insight: thee phenomenologically distinct modes of decision making.* PhD Thesis, University of Alabama.

Harper P and Pitt M (2004). On the challenges of healthcare modelling and a proposed project life cycle for successful implementation. *Journal of the Operational Research Society* **55(6)**: 657-661.

Hélie S and Sun R (2010). Incubation, insight, and creative problem solving: a unified theory and a connectionist model. *Psychological Review* **117(3)**: 994-1024.

Jones G (2003). Testing two cognitive theories of insight. *Journal of Experimental Psychology Learning, Memory & Cognition* **29(5)**:1017.

Kahneman D (2011). *Thinking, fast and slow*. Farrar, Straus and Giroux: New York, NY, US.

Kaplan C A and Simon H A (1990). In search of insight. *Cognitive Psychology* **22(3)**: 374-419.

Kershaw T C and Ohlsson S (2004). Multiple causes of difficulty in insight: the case of the nine-dot problem. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **30(1)**: 3-13.

Kopainsky B, Alessi S M, Pedercini M and Davidsen P I (2009). Exploratory strategies for simulation-based learning about national development. In: Ford A, Ford D N and Anderson E G (eds). *Proceedings of the 27th International Conference of the System Dynamics Society*. System Dynamics Society: New Mexico, USA.

Kotiadis K, Tako A A and Vasilakis C (2013). A participative and facilitative conceptual modelling framework for discrete event simulation studies in healthcare. *Journal of the Operational Research Society* advance online publication 6 March, doi:10.1057/jors.2012.176.

Kounios J and Beeman M (2009). The Aha! moment the cognitive neuroscience of insight. *Current Directions in Psychological Science* **18(4)**: 210-216.

Kounios J, Fleck J I, Green D L, Payne L, Stevenson J L, Bowden E M and Jung-Beeman M (2008). The origins of insight in resting-state brain activity. *Neuropsychologia* **46(1)**: 281-291.

Kounios J, Frymiare J L, Bowden E M, Fleck J I, Subramaniam K, Parrish T B and Jung-Beeman M (2006). The prepared mind: neural activity prior to problem presentation predicts subsequent solution by sudden insight. *Psychological Science* **17(10)**: 882-890.

Langley P A and Morecroft J D (2004). Performance and learning in a simulation of oil industry dynamics. *European Journal of Operational Research* **155(3)**: 715-732.

Lee C B (2010). The interactions between problem solving and conceptual change: System dynamic modelling as a platform for learning. *Computers & Education* **55**: 1145-1158.

Maier N R F (1940). The behavior mechanisms concerned with problem solving. *Psychological Review* **47(1)**: 43-58.

Mayer R E (2010). Problem solving and reasoning. In: Peterson P, Baker E and McGaw B (eds). *International Encyclopedia of Education*. 3rd Edition. Elsevier: Oxford, pp 273-278.
Mayer R E (1995). The search for insight: grappling with Gestalt psychology's unanswered questions. In: Sternberg R J and Davidson J E (eds). *The Nature of Insight*. The MIT Press: Cambridge, MA, US, pp 3-32.
Metcalfe J and Wiebe D (1987). Intuition in insight and noninsight problem solving. *Memory & Cognition* **15(3)**: 238-246.
Monks T, Robinson S and Kotiadis K (2014). Learning from discrete-event simulation: exploring the high involvement hypothesis. *European Journal of Operational Research*, forthcoming.
Monxes E (2004). Misperceptions of basic dynamics: the case of renewable resource management. *System Dynamics Review* **20(2)**: 139-162.
Morecroft J D (1992). Executive knowledge, models and learning. *European Journal of Operational Research* **59 (1)**: 9-27.
Öllinger M, Jones G and Knoblich G (2008). Investigating the effect of mental set on insight problem solving. *Experimental Psychology* **55(4)**: 270-282.
Öllinger M, Jones G, Faber A H and Knoblich G (2013). Cognitive mechanisms of insight: the role of heuristics and representational change in solving the eight-coin problem. *Journal of Experimental Psychology: Learning, Memory, and Cognition* **39(3)**: 931-939.
Paul R J and Kuljis J (2010). Problem solving, model solving, or what?. In: Johansson B, Jain S, Montoya-Torres J, Hugan J and Yücesan E (eds). *Proceedings of the 2010 Winter Simulation Conference*. Baltimore, USA, pp 353-358.
Robinson S (2001). Soft with a hard centre: discrete-event simulation in facilitation. *Journal of the Operational Research Society* **52**: 905-915.
Robinson S (2004). *Simulation: the practice of model development and use*. John Wiley and Sons Ltd: Chichester.
Robinson S (2005). Discrete-event simulation: from the pioneers to the present, what next?. *Journal of the Operational Research Society* **56**: 619-629.
Robinson S and Pidd M (1998). Provider and customer expectations of successful simulation projects. *Journal of the Operational Research Society* **49**: 200-209.
Rouwette E A J A, Korzilius H, Vennix J A M and Jacobs E (2011). Modeling as persuasion: the impact of group model building on attitudes and behavior. *System Dynamics Review* **27(1)**: 1-21.
Schooler J W, Fallshore M and Fiore S M (1995). Epilogue: putting insight into perspective. Sternberg R J and Davidson J E (eds). *The Nature of Insight*. The MIT Press: Cambridge, MA, US, pp 559-587.
Seifert C M, Meyer D E, Davidson N, Patalano A L and Yaniv I (1995). Demystification of cognitive insight: Opportunistic assimilation and the prepared-mind hypothesis. In: Sternberg R J and Davidson J E (eds). *The Nature of Insight*. The MIT Press: Cambridge, MA, US, pp 65-124.
Senge P M and Sterman J D (1992). Systems thinking and organizational learning: Acting locally and thinking globally in the organization of the future. *European Journal of Operational Research* **59(1)**: 137-150.
Simon H A (1979). *Models of thought*. Yale University Press: Binghampton, NY, US.
SIMUL8 Corporation (2013). Simul8 Simulation Software.
Smith D (1990). The use of microcomputer-based simulation models in the teaching of operations management. *International Journal of Operations & Production Management* **10(5)**: 5-14.
Smith S M (1995). Getting into and out of mental ruts: a theory of fixation, incubation, and insight. In: Sternberg R J and Davidson J E (eds). *The Nature of Insight*. The MIT Press: Cambridge, MA, US, pp 3-32.
Sterman J D (1994). Learning in and about complex systems. *System Dynamics Review*, **10(2-3)**: 291-330.
Sterman J D (1989a). Misperceptions of feedback in dynamic decision making. *Organizational Behavior and Human Decision Processes* **43(3)**: 301-335.
Sterman J D (1989b). Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment. *Management Science* **35(3)**: 321-339.
Sternberg R J (2009). *Cognitive Psychology*. 5th Edition, Cengage Learning: Wadsworth.

Sternberg R J (1999). *Handbook of creativity*. Cambridge University Press: New York.

Topolinski S and Reber R (2010). Gaining insight into the 'aha' experience. *Current Directions in Psychological Science* **19(6)**: 402-405.

van der Zee D and Slomp J (2009). Simulation as a tool for gaming and training in operations management - a case study. *Journal of Simulation* **3(1)**: 17-28.

Ward T B, Smith S M and Finke R A (1999). Creative cognition. In: Sternberg R J (ed)., *Handbook of Creativity*. Cambridge University Press: New York, pp 189-212.

Warren K and Langley P (1999). The effective communication of system dynamics to improve insight and learning in management education. *Journal of the Operational Research Society*, **50(4)**: 396-404.

Weisberg R W (1995). Prolegomena to theories of insight in problem solving: a taxonomy of problems. In: Sternberg R J and Davidson J E (eds). *The Nature of Insight*. The MIT Press: Cambridge, MA, US, pp 157-196.

Weisberg R W (1999). Creativity and knowledge: a challenge to theories. In: Sternberg R J (ed). *Handbook of Creativity*. Cambridge University Press: New York, NY, USA, pp 226-250.

Wynder M (2004). Facilitating creativity in management accounting: a computerized business simulation. *Accounting Education* **13(2)**: 231-250.

AUTHOR BIOGRAPHIES

ANASTASIA GOGI is a PhD student at the School of Business and Economics, Loughborough University. Anastasia holds a BSc in Industrial Management (University of Piraeus, Greece) and an MSc in Management Sciences (University of Southampton, UK). The aim of her current research is to determine and provide empirical research on how the use of simulation models leads people to generate insights, working closely with SIMUL8 Corporation as the industrial partner. Prior to her current studies, she worked as a business analyst for a shipping company.

ANTUELA A. TAKO is a Lecturer in Operations Research at the School of Business and Economics, Loughborough University. She holds a PhD in Simulation and an MSc in Management Science and Operational Research from the University of Warwick. She previously worked for a research project that introduces stakeholder participation and facilitation in discrete-event simulation modelling. Her research interests include the comparison of simulation approaches (discrete-event simulation and system dynamics), participative simulation modelling and conceptual modelling. Home page: http://info.lut.ac.uk/departments/sbe/staff/academic-research/bsat2.html

STEWART ROBINSON is Professor of Management Science and Associate Dean Research at Loughborough University, School of Business and Economics. Previously employed in simulation consultancy, he supported the use of simulation in companies throughout Europe and the rest of the world. He is author/co-author of five books on simulation. His research focuses on the practice of simulation model development and use. Key areas of interest are conceptual modelling, model validation, output analysis and alternative simulation methods (discrete-event, system dynamics and agent based). Professor Robinson is co-founder of the Journal of Simulation and President of the Operational Research Society. Home page: www.stewartrobinson.co.uk

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

GRAPHICAL REPRESENTATION OF AGENT-BASED MODELS IN OPERATIONAL RESEARCH AND MANAGEMENT SCIENCE USING UML

Peer-Olaf Siebers

School of Computer Science
University of Nottingham
Nottingham, NG8 1BB, United Kingdom
peer-olaf.siebers@nottingham.ac.uk

Bhakti S. S. Onggo

Department of Management Science
Lancaster University Management School
Lancaster, LA1 4YX, United Kingdom
s.onggo@lancaster.ac.uk

ABSTRACT

Agent-Based Modelling and Simulation (ABM/S) is still struggling to become one of the main stream simulation methods in Operational Research (OR) and Management Science (MS), despite its generally accepted usefulness when it comes to representing human behaviour in human-centric systems. In other fields, as for example Business Studies, Economics, and Social Science, it is flourishing. One of the technical differences between ABM/S and the well-established OR/MS simulation methods System Dynamics Simulation (SDS) and Discrete Event Simulation (DES) is that ABM/S traditionally uses an equation based modelling approach while SDS and DES use a graphical notation for the model description. We believe that having a graphical notation for ABM/S would help establish it in OR/MS. The Unified Modelling Language (UML) is a graphical notation commonly used in software engineering for the purpose of software design. Use case and state machine diagrams, which are part of the UML notation seem to lend themselves particularly well to ABM/S. In this paper we introduce UML to the OR/MS community. First we explain step-by-step how to use UML for developing ABM/S models. Then we demonstrate the application of this graphical notation by presenting two conceptual models we built for real world OR/MS case studies.

Keywords: Operational Research, Management Science, agent-based modelling, agent-based simulation, UML, model notation, conceptual modelling, use case diagram, state machine diagram, state chart

1 INTRODUCTION

The topic of the panel discussion at the 2010 Operational Research Society Simulation Workshop was the relevance of Agent-Based Modelling and Simulation (ABM/S) in Operational Research (OR) and Management Science (MS)[1]. The discussion highlighted that representing human behaviour within operations and service systems with the help of ABM/S was still in its infancy in OR (Siebers et al 2010). We were interested to see if anything had changed over the last 4 years. A quick search in the *International Abstracts in Operations Research* (IAOR) database for two 4 year periods using keywords related to the topic of OR simulation[2] shows that the reported use of ABM/S has nearly doubled within the last 4 years (Table 1). But we need to interpret this result with care. Spot checks uncovered that most of these papers relate to using ABM for optimising systems rather than for representing human behaviour within operations and service systems. When searching for "social simulation" we found that this was a hot topic in OR but there were only very few papers that mentioned "agent based" in their keyword list. To our surprise we did not find even one paper that mentioned "agent based" in conjunction with UML, a graphical notation that is used to define Discrete Event Simulation (DES) models, in its keyword list.

---

[1] For simplicity, from now on we will only refer to OR, but mean in fact OR/MS
[2] For a brief description of the different simulation paradigms see Siebers and Aickelin (2008)

Table 1 *IAOR search results*

Term 1	Term 2	2006-2009	2010-2013
Simulation		1298	2049
Simulation	System dynamics	73	128
Simulation	Discrete event	119	93
Simulation	Agent based	47	85
Simulation	UML	2	5
Agent based	UML	0	0
Social simulation		38	83
Social simulation	Agent based	3	12

In summary, there seem to be some recognition of ABM/S among the OR community but its usage for modelling human behaviour within operations and service systems has not reached the desired extent as suggested in Siebers et al (2010). So what is stopping the OR community from using ABM/S for their purposes? Perhaps it is the lack of "knowing how to" use it. As ABM/S has been widely adopted in in related fields like Business Studies, Economics, or the Social Sciences (Macal and North 2010), the obvious solution would be to adopt well-established approaches from one of the other disciplines. But there are some stumbling blocks that prevent us from doing so. If we consider the different goals we are trying to achieve as well as the different foundations on which we base our models, we can quickly see that it is not a straightforward process to simply adopt an established approach from Business Studies, Economics, or the Social Sciences. Table 2 provides an overview of the differences between OR and Business Studies, Economics, and the Social Sciences in the way they approach simulation modelling and in the way they use simulation models.

Table 2 *Different approaches to simulation modelling(adapted from Robinson 2010)*

Operational Research	Business, Economics and Social Science
Focus on solving real world problems	Focus on understanding the real world
Data-driven (empirical work)	Theory-driven
Good quality of data and strong analysis	Sound dynamics hypotheses
Validation to ensure sufficient accuracy for purpose	Plausibility rather than operational validity[3]
Implementation of findings	Learning and understanding

How can we make OR ABM/S more accessible and attractive? One problem that we think limits the success of ABM/S in OR is that the other disciplines that use ABMs for modelling social systems use an equation based approach to create their models which is not compatible with the way we generally do simulation modelling in OR. Ever since the proliferation of visual modelling, we have been using graphical notations in many simulation projects (stock and flow diagrams in System Dynamics Simulation (SDS) and process flow diagram in DES). The extensive use of graphical notations can be seen from many simulation modelling courses offered by OR departments. Therefore our hypothesis is that if a graphical notation can be established, the number of users of OR ABM/S will grow more rapidly. A graphical notation that could be used for ABM/S includes *Unified Modeling Language* (UML) (OMG 2010b) and *Business Process Model and Notation* (BPMN) (OMG 2010a). In this paper, we focus on UML since BPMN has been explained in Onggo (2012, 2013). Our current work uses the standard UML (not adding any proprietary extension).

UML has been used in the field of Computer Science but is not well established in OR. The use of a graphical notation makes it easier for the stakeholders who are not an expert in Mathematics to understand and validate the models that are created. This in turn will improve the credibility of the models. So it is important to convince OR practitioners about the usefulness of UML as a modelling

[3] For an explanation on "operational validity" see Sargent (2005)

tool. There is also a need to provide some UML training for members of the OR community in order to spread the use of OR ABM/S.

Another problem that we think limits the success of ABM/S in OR is that it is often assumed that agent models require enormous computational power. While this is true for some models it does not apply to all models. If the right level of abstraction is chosen this problem might become less severe. For example, rather than building a model with 1,000,000 complex agents one could consider building a large scale model with 1,000,000 simple agents to capture population dynamics and several small scale models with 1,000 complex agents to study the impact of specific events on human behaviour. Alternatively, population dynamics could be captured by using SDS while the impact of specific events on human behaviour could be studied by using ABM/S. Therefore our hypothesis is that if such considerations are taken into account, ABM/S becomes a feasible modelling option and the use of ABMs could grow more rapidly.

The objective of this paper is to demonstrate how UML diagrams can be used to represent OR agents. We aim to demonstrate that given the right simulation development tool, using ABM/S is not more difficult than using other simulation techniques. The remainder of this paper is organised as follows: Section 2 introduces the notation of UML state machine diagrams and provides a step-by-step guide to developing UML state machine diagrams for OR. Section 3 provides an overview of potential tools to be used for ABM/S in OR. In Section 4 we demonstrate the application of UML state machine diagrams by presenting two conceptual models we built for real world OR case studies. Section 5 provides conclusions and an outlook into the future of OR ABM/S.

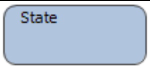
2 THE APPLICATION OF UML TO OR ABM/S

The UML is a family of graphical notations that is used in the field of software engineering for describing and designing object oriented software systems (Fowler 2004). The latest UML standard (v2.4) comprises 26 different diagram types. While some of these types are used for capturing the structural design of a software system others are used for capturing the behavioural design of a software system. Often programmers will only need a proper subset of the existing diagrams in order to express the semantics of a large percentage of analysis and design issues (Booch et al 2007). Perhaps the most useful diagrams for ABM/S in OR are use case diagrams, class diagrams, sequence diagrams, and state machine diagrams. A *use case diagram* is commonly used for the communication between the key stakeholders. They depict who (or what) interacts with the system. A *class diagram* describes the types of objects (agents) in the system and the various kinds of static relationships between objects. A class can be seen as a template for the agents that will be created. A *sequence diagram* describes how groups of objects collaborate with each other (by showing the messages passed between the objects) in a specific scenario (use case). A *state machine diagram* (also called state chart) describes the lifetime behaviour of a single object. They express behaviour as a progression through a serious of states, triggered by events, and the related actions that might occur. These diagrams can be very effective in visually capturing the logic within agents and quickly conveying the underlying dynamics of complex models (Ozik et al 2013). In this paper we focus on state chart development although for the case studies (section 4) we will also present some use case diagrams to support the understanding of the state charts.

2.1 UML State Charts

In ABM/S, a state chart can be used to represent the behaviour of an agent at some discrete points in time. An agent will move from one state to another and during the transit, an agent may execute an action. A state chart diagram has a number of graphical elements. The main elements are shown in Table 3.

Table 3 *State chart's main graphical elements*

Graphical element	Description
	Entry pointer: Indicates the initial state after an object is created
	State: Represents a locus of control with a particular set of reactions to conditions and/or events
	Initial state pointer: Points to the initial state within a composite state
	Final state: Termination point of a state chart
	Transition: Movement between states, triggered by a specific event
	Branch: Transition branching and/or connection point
	Shallow history: The state chart remembers the most recent active sub state (but not the lower level sub-states)
	Deep history: The state chart remembers the most recent active sub state (including the lower level sub states)

A simple example of a state chart diagram can be found in Figure 1. A state chart has exactly one *state chart entry pointer* which indicates the initial state of the agent. A *state* models a situation during which some invariant condition holds. Usually time is consumed while an agent is in a specific state. A *simple state* is a state that does not have substates while a *composite state* is a state that has substates (nested states). A state chart has as many *initial state pointers* as it has *composite states*.

In this example, at the top level, the state chart has two *composite states*. The state on the left has further two substates. It also has a *shallow history state*. This means after the state chart control has moved from the left composite state to the right composite state, the next time the control moves back to the left composite state, it will remember the last substate it was in before leaving the composite state and will directly jump to that substate. The composite state on the right has one composite substate and one simple substate. It also has a *deep history state*. The deep history state allows the composite substate to remember the most recent active states including the lower level substates within it. If we would replace the deep history state with a shallow history state, then the composite substate would not be able to remember the simple state it was in before leaving the composite state. Instead it will use the default state pointed to by the initial state pointer. A *transition* indicates that if the specified trigger event occurs and the specified guard condition is true, the control within the state chart moves from one state to another and performs the specified action (if one was specified). In a state chart it is also possible to define *branches*, i.e. decision points that determine which transitions to follow next. A *final state* is the termination point of a state chart. When the control enters a final state, its action is executed, and the state chart terminates. In technical terms this means that the agent object is deleted from computer memory. A more in depth introduction to the UML state chart elements can be found in Fowler (2004) or Booch et al (2007).

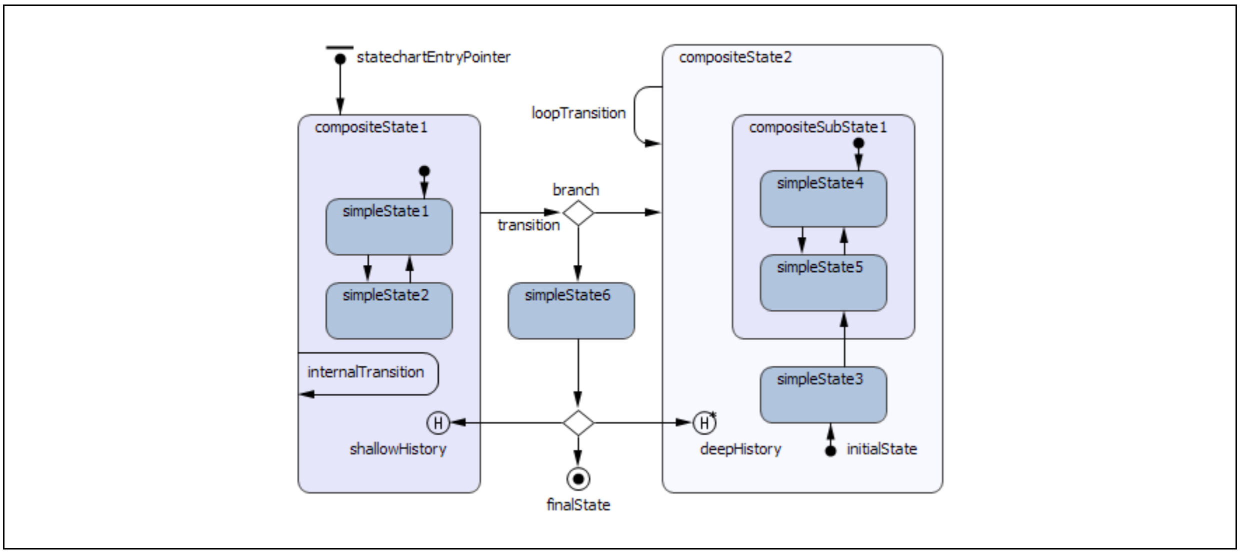


Figure 1 *Typical elements of a state chart diagram that can be used to represent an agent*

2.2 Common OR ABM Model Patterns

Modellers may ask if there are some standard design patterns for ABMs. As such there are no standard design patterns. But there are typical designs that might help you to get going in the right direction and there are designs that often get you going in the wrong direction. Based on our experience, we have seen three common design patterns: centralised design (Figure 2, left side), decentralised design (Figure 2, right side), and hierarchical design (not depicted). The latter is often used in combination with the former design patterns. In a centralised design there is one centre state that one will return to before getting into any other state or loop of states. This state can be thought of as a "contemplating" state that normally does not consume any time. It is a dummy state that is supposed to help simplifying the design of the state chart. But the centre state can also be one that consumes time (e.g. a specific location from which all activities start). An alternative design is the decentralised design where the central state is missing. State charts can also be hierarchical, i.e. each state can contain another state chart, making that state a composite state.

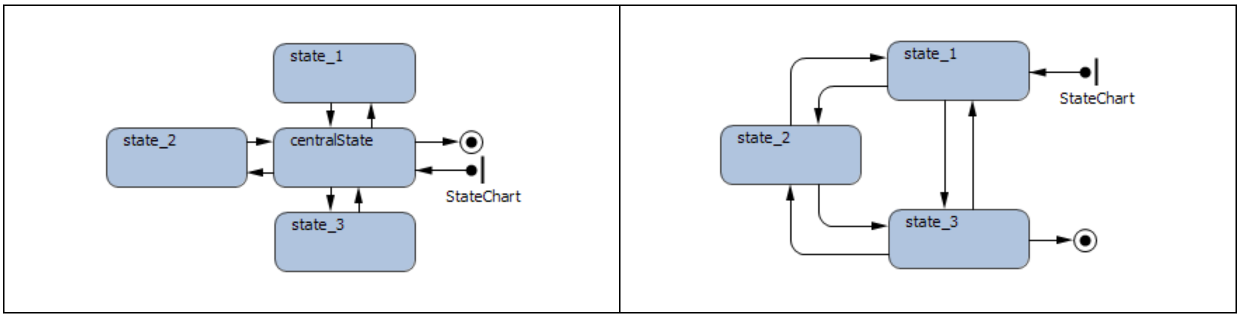


Figure 2 *Typical designs (left: centralised; right: decentralised)*

2.3 A Step-By-Step Example of Building an OR ABM

There are many different strategies of getting started building ABMs and it is impossible (and perhaps not very useful) to list them all here. Below, we describe a strategy that often has helped us to get started with designing our OR ABMs. As an example case, we have chosen to model office workers.

The development starts with thinking about what actors we want to consider in our models. Once we know the actors, in our example "office workers", we need to define the states they can be in. In order to come up with potential states it helps to think in terms of "locations" first. For an office worker relevant locations might be: "at home", "at the office", or "elsewhere". The next step would be to think about key "time consuming activities" within these locations. It is important to consider only key locations and key activities as otherwise the state chart gets too complex. One should only define as much detail as is really necessary for investigating the question studied. In our example details of what the worker is doing at home are not really relevant to the study (which focuses on work habits) and therefore the "atHome" state is a single state only, measuring the time the worker spends at home. But the details of the "atOffice" state are relevant to the study (to measure the time the worker is actually working). Therefore, the "atWork" state is modelled as a composite state that contains a sub statechart differentiating the time spent for "working" and "dozing". The result of this modelling process could look like the left side of Figure 3. It shows a centralised design of a worker agent with "atHome" being the centre state. If the modeller wants to consider that the worker should leave the job and go elsewhere without going home and going from elsewhere directly to work transitions between "elseWhere" and "atOffice" could be added which would make this a decentralised design (as shown on the right side of Figure 3). Once the graphical design of the state chart is finished the modeller has to consider how state changes will be triggered. These could be triggered by conditions, timeouts, rates, or schedules. Finally the modeller needs to consider how the agents will interact, i.e. what kind of messages should be sent and who should receive them and in which situation. Once this conceptual modelling process is finished and the design has been validated by the stakeholders, the implementation can begin. It is important to remember that the whole modelling process described above is an iterative process and that it might take a while to come up with the final design of an agent's state chart.

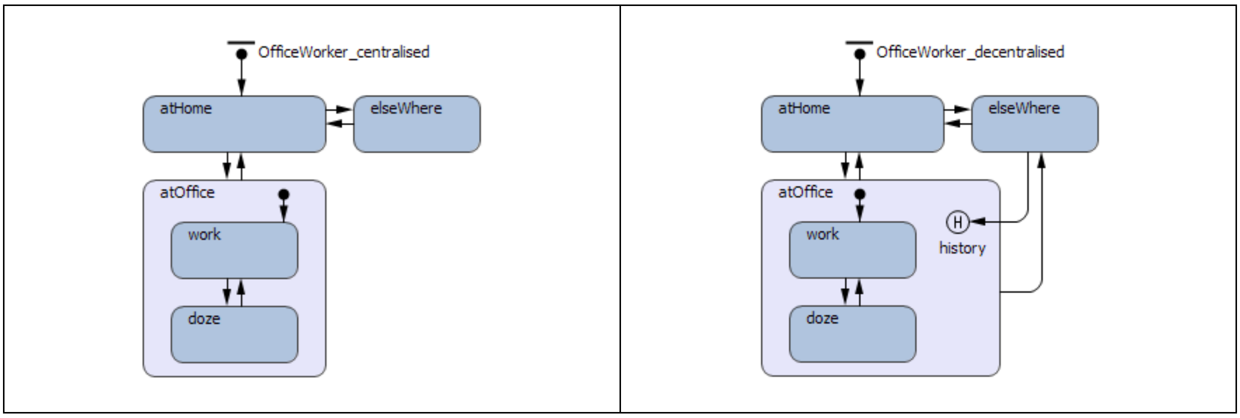


Figure 3 *Basic state chart of office workers (left: centralised design; right: decentralised design)*

3 UML ABM/S TOOLS

Nowadays there are several simulation modelling tools that support the use of UML state charts as a graphical notation for the modelling process and these tool can automatically translate the graphs into executable code.

Perhaps the most advanced tool is the commercial software AnyLogic (http://www.anylogic.com/) which is a multi-paradigm eclipse based simulation IDE that supports graphical model design for all major simulation paradigms (SDS, DES, and ABM/S). When it comes to the implementation stage, a modeller who plans to use AnyLogic needs to have some knowledge of Java. When a model gets more complex it often requires some programming (e.g. to define a more complex behaviour triggered by an event or to define a more complex transition rules). A very useful feature of AnyLogic is the support of hierarchical model design (e.g. a state chart inside another state chart) and hybrid model design (e.g. SDS and DES sub models in a single multi-paradigm model). Hybrid models can also be hierarchical. An alternative is Repast Simphony (http://repast.sourceforge.net/). It is a free and open source ABM/S platform. In its latest version, it has been equipped with a new agent state chart framework (Ozik et al 2013). Repast Simphony is a java-based modelling system that is designed for use on workstations. A lean and expert-focused C++ based modelling system that is designed for use on large computing clusters and supercomputers is also available but does not support graphical agent modelling (however, it supports a Logo-like programming language which is designed to be simpler than C++). The most commonly used ABM/S tool in academia is NetLogo (http://ccl.northwestern.edu/netlogo/). While it does not support UML (and is in fact not object oriented) it has the advantage that it is easy to learn, as it uses a simple programming language based on Logo. Furthermore many sample models and tutorials exist for this tool, much more than for the other tools mentioned above. There are many other tools but only very few of these support UML based modelling. An up-to-date list of existing ABM/S tools is provided at Wikipedia (2013b).

4 CASE STUDIES

In this section we present two conceptual models that we have developed for real world case studies. Here we focus on the graphical representation rather than on the case studies themselves. Detailed information about the associated case study investigations can be found in Siebers and Aickelin (2011) and Zhang et al (2010). In order to support the UML state chart description we also provide some UML use case diagrams which are commonly used for the communication between the key stakeholders. They depict who (or what) interacts with the system. In these diagram *actors* are entities that interface with the system while *use cases* represent what the actors want the system to do for them. The associations in the use case diagram indicate which actors initiate which use cases.

4.1 Case Study 1: Simulating People Management Practices in Retail

In the first case study, we use a combination between ABM and DES to understand the impact of management practices on company performance in a retail environment. In the retail sector (e.g. department store operations) operational management practices are very well researched while people management practices are often neglected by the researchers. The problem seems to be that the tool usually used for such investigations (DES) does not well support the studies of people management practices. In our studies we proposed to use a combination between ABM and DES to model this service system as we have a human centric system with an underlying queuing structure. We use a queuing system for modelling the operations (queues and service priorities within the department store) while we use agents for modelling staff and customers and their behaviours. These agents replace the passive entities traditionally used within the DES system. This is an improvement compared to using only DES as it allows us to model real world human behaviour and consider things like the evolution of customer preferences over time. Also pure DES would not allow us to consider proactive behaviour while ABM/S supports this concept. To demonstrate the usefulness of such an approach we conducted a case study with a top ten UK retailer to empirically inform the modelling and simulation process looking at Audio & TV (A&TV) and Womenswear (WW) departments across two branches.

We started our case study with knowledge gathering through informal participant observations, staff interviews, and informational sources internal to the case study organisation. Then we continued with the conceptual modeling. We decided to consider two types of agents: customers and staff (while staff agents were further subdivided into different types). All staff agents shared a standard design. Figure 4 shows an overview of the conceptual model with customer and staff agents. The transitions are triggered by using frequency distributions (e.g. triangular distributions for determining state change delays) and the decision making was represented using probability distributions. We also considered different types of customers (with different likelihoods to engage in certain activities and different patient levels) by implementing several archetypes for our customer agents. This gave us some control over the behaviour of our customer agents. Using the state chart approach for the conceptual modelling also helped us to communicate the model to our stakeholders (management staff at different levels at the case study department store) and supported very well the implementation of this model. For the implementation we used AnyLogic. The archetyping together with the object oriented modelling approach allows us to collect statistics about different types of customers. Our agents were also fitted with a novel customer satisfaction measure which evaluated customer satisfaction at each state transition. Without modelling the customers as agents implementing such a measure would have not been possible.

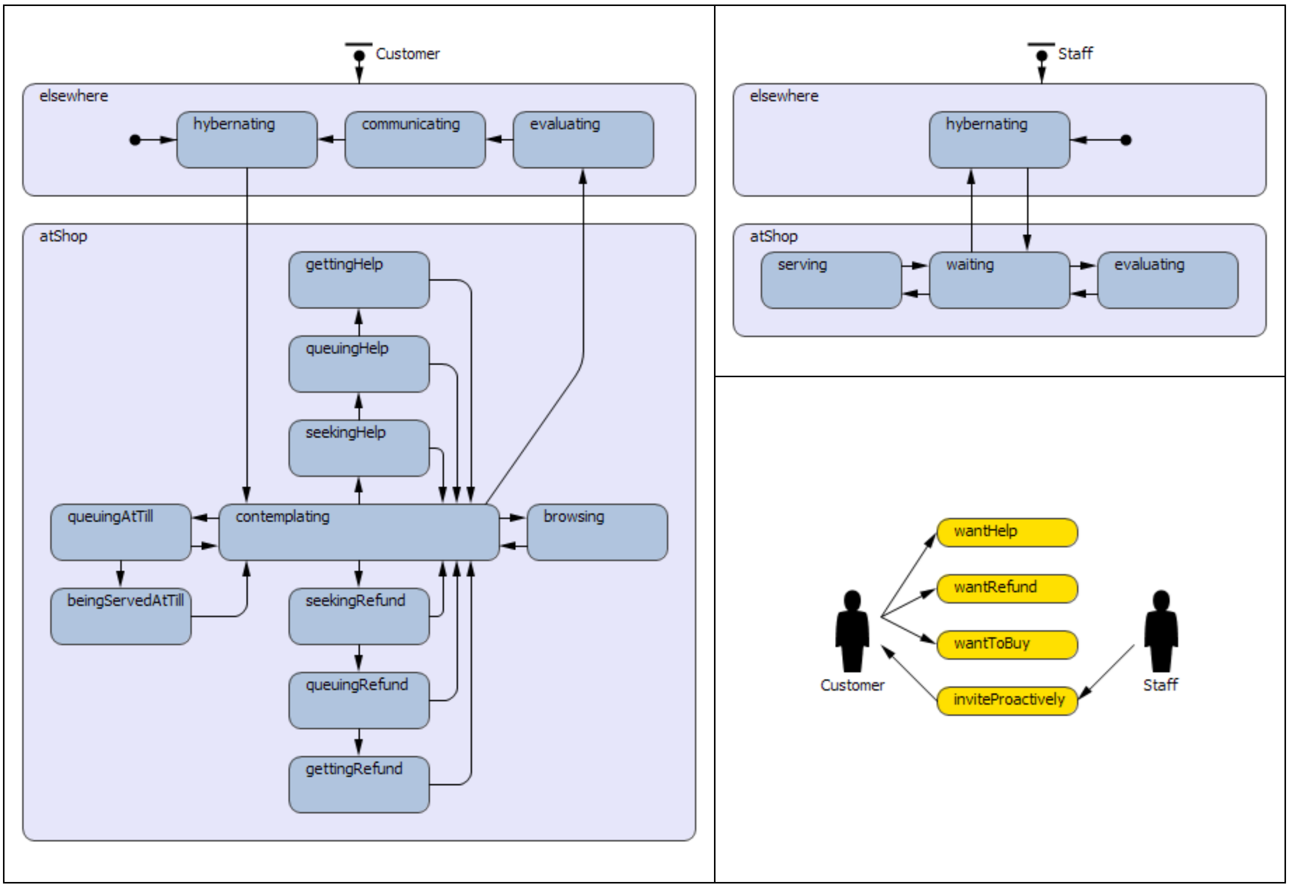


Figure 4 *Agent state charts (left: customer; top right: staff) and use case diagram (bottom right)*

4.2 Case Study 2: Office Building Energy Consumption

In the second case study we use ABM to investigate "Office Building Energy Consumption". Each organisation faces a dilemma in terms of energy consumption. It has to provide energy to satisfactorily meet the energy needs of staff and maintain comfort standards in its office buildings but it also has to minimise its energy consumption through effective energy management policies in order to reduce energy bills. Our goal in this case study was to demonstrate the applicability of ABM/S for testing the effectiveness of different energy management strategies. To keep things simple we focused on the electricity consumption. As a case study system we chose our university building. As there is no notion of queues in this model we used a pure ABM/S approach. The purpose of the model was to provide university estate managers with a decision support tool to help them with decisions like: shall we use automated or manual lighting management.

In order to gather some knowledge we consulted with the school's director of operations and the university estate office. We also conducted a survey amongst the school's 200 PhD students and staff on electricity use behaviour. From the survey results we were able to identify some archetypes regarding working hour habits (early birds; timetable compliers; flexible workers) and energy saving awareness (environment champion; energy saver; regular user; big user). We decided to consider four different types of agents: energy user (including the PhD students and staff), computers, lights, and offices. While energy users are considered as active entities the other three types of agents are considered as passive entities. Figure 5 shows an overview of the main statechart diagrams. For the design of the energy user agent we followed the design principles of the worker agent mentioned in Section 2.3. First the location states are defined and then time consuming activities are defined within the location states.

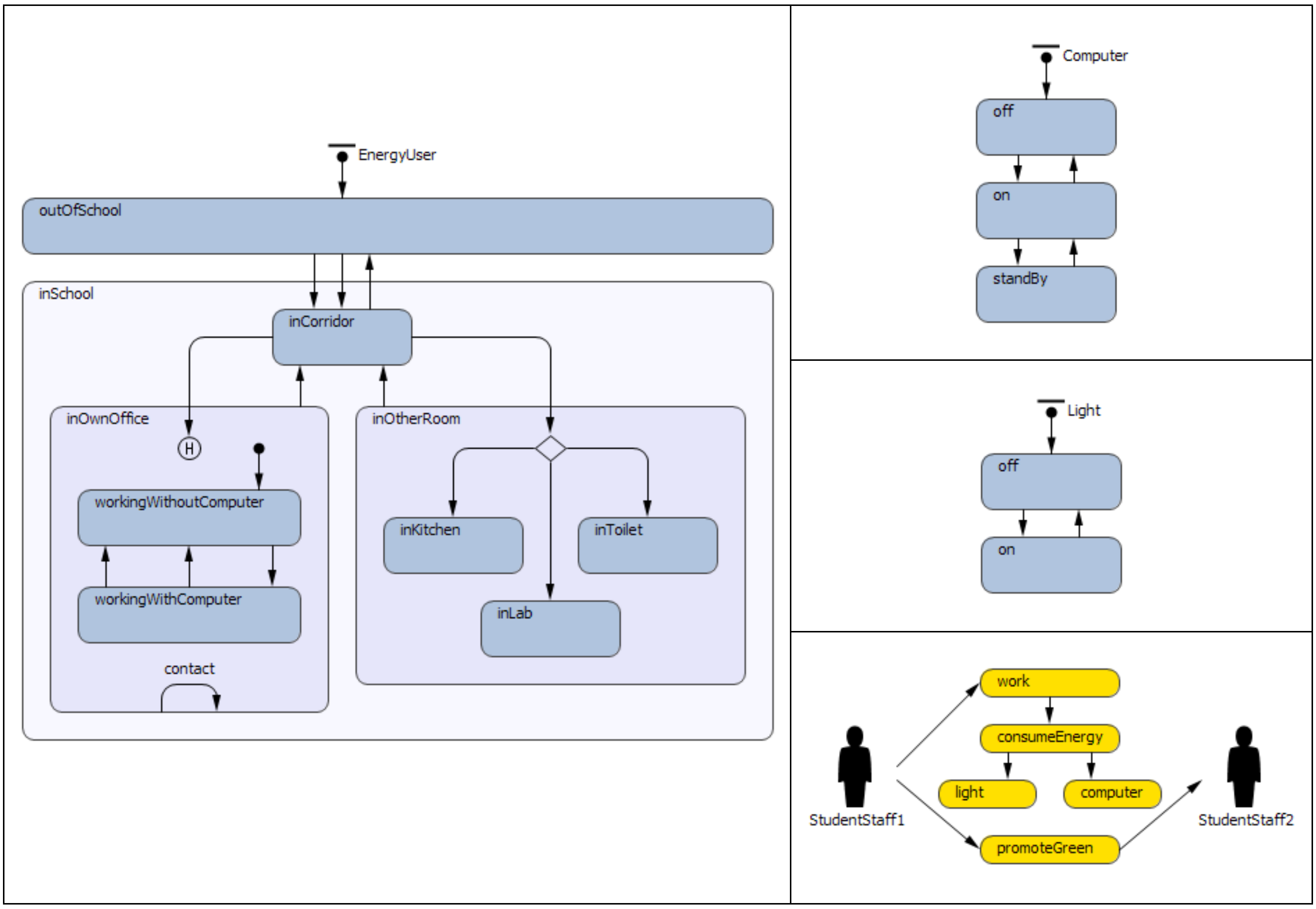


Figure 5 *Agent state charts (left: energy user; top right: computer, light) and use case diagram (bottom right)*

5 CONCLUSION

In this paper we have demonstrated how the use of a graphical notation can simplify the life of an OR practitioner (both, academic and industry) who want to enjoy the benefits of using ABM/S. We strongly believe that ABM/S has the potential to become the predominate paradigm for modelling human behaviour in human-centred systems in OR in the near future. Suitable tools do already exist but what is missing is the promotion of these tools in the OR community (see also Siebers et al 2010). Training should be organised to enable people to take advantage of this technology.

Our next step will be to look at other relevant UML diagrams (use case diagram, class diagram, and sequence diagram) as they will allow us to provide a more complete specification of our agents, their interactions, and the environment in which they live in. This should then help to improve the conceptual modelling process (for example, to communicate the model to people who may not be familiar with computer codes) and the validation of the models as it allows us to better explain the structure and behaviour of our simulation models. The co-author has also worked into the use of Business Process Modelling Notation (BPMN) in representing ABMs. This notation is similar to the UML state chart notation but might be more familiar to people from Business Schools and therefore easier to use for OR studies. Unfortunately, there is no ABM/S tool that supports BPMN at the moment. However, a prototype has been developed to show that this is possible (Onggo and Karpat 2011).

ACKNOWLEDGMENT

We would like to thank Gaurang Phadke (Business School, Bayreuth University, Germany) and Holger Schnädelbach (School of Computer Science, Nottingham University, United Kingdom) for their help and we also very much appreciate the valuable comments from the reviewer.

REFERENCES

Booch G, Maksimchuk RA, Engle MW, Young BJ, Conallen J and Houston KA (2007) Object-Oriented Analysis and Design with Applications, 3rd Edition, Boston, MA: Pearson Education

Fowler (2004) UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition, Addison Wesley.

Macal CM and North MJ (2010). Tutorial on agent-based modelling and simulation. Journal of Simulation, 4(3), 151-162.

Object Management Group (2010a). Business Process Model and Notation (BPMN) version 2.0. [online] Available from http://www.bpmn.org [Accessed Nov 15, 2013].

Object Management Group (2010b). Object Management Group Unified Modeling Language (OMG UML), Superstructure version 2.4. [online] Available from http://www.omg.org [Accessed Nov 15, 2013]

Onggo B S S and Karpat O (2011). Agent-based conceptual model representation using BPMN. *Proceedings of the 2011 Winter Simulation Conference*, 11-14 December 2011, Phoenix, Arizona (WSC11). Los Alamitos, California: IEEE Computer Society Press, pp. 671-682.

Onggo B S S (2012). BPMN Pattern for Agent-Based Simulation Model Representation. *Proceedings of the 2012 Winter Simulation Conference*, 9-12 December 2012, Berlin, Germany (WSC12). Los Alamitos, California: IEEE Computer Society Press.

Onggo B S S (2013). Agent-Based Simulation Model Representation Using BPMN. In: Fonseca, P. (Ed) *Formal Languages for Computer Simulation: Transdisciplinary Models and Applications*. Hershey, PA: Information Science Reference, pp. 378-399.

Ozik J, Collier N, Repast Development Team (2013). Repast Statecharts Guide. [online] Available from http://repast.sourceforge.net/docs/Statecharts.pdf. [Accessed Nov 15, 2013].

Robinson S (2010) Modelling Service Operations: A Mixed Discrete-Event and Agent Based Simulation Approach. Talk given at the Western OR Discussion Society on 27 April 2010

Sargent R (2005). Verification and validation of simulation models. Proceedings of the 37th Winter Simulation Conference. IEEE, Inc., 2005, pp. 130–143.

Siebers PO, Macal CM, Garnett J, Buxton D and Pidd M (2010). Discrete-Event Simulation is Dead, Long Live Agent-Based Simulation!. *Journal of Simulation*, 4(3) pp. 204-210.

Siebers PO and Aickelin U (2008). Introduction to Multi-Agent Simulation. In: Adam F and Humphreys P (Eds). Encyclopedia of Decision Making and Decision Support Technologies, Pennsylvania: Idea Group Publishing, pp. 554-564.

Siebers PO and Aickelin U (2011). A First Approach on Modelling Staff Proactiveness in Retail Simulation Models. *Journal of Artificial Societies and Social Simulation*, 14(2) 2.

Zhang T, Siebers PO and Aickelin U (2011). Modelling Electricity Consumption in Office Buildings: An Agent Based Approach. *Energy and Buildings*, 43(10) pp. 2882-2892.

AUTHOR BIOGRAPHIES

PEER-OLAF SIEBERS is a lecturer in the School of Computer Science at the University of Nottingham in the United Kingdom (http://www.cs.nott.ac.uk/~pos/). His main research interest is the application of data driven computer simulation to study human-centric complex adaptive systems. This is a highly interdisciplinary research field, involving disciplines like social science, psychology, management science, Operational research, economics and engineering. Other areas of interest include Risk Analysis and Systems Biology.

BHAKTI S S ONGGO is a lecturer in the Department of Management Science at the Lancaster University Management School, Lancaster, United Kingdom. His research interests lie in the areas of simulation methodology (modelling paradigms and conceptual modelling), simulation technology (parallel and distributed simulation, web-based simulation cloud-based simulation), business process modelling and simulation applications. His email address is s.onggo@lancaster.ac.uk.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

CONCEPTUAL MODELLING: LESSONS FROM COMPUTER SCIENCE

Fahim Ahmed

School of Business and Economics
Loughborough University
Loughborough, LE11 3TU
United Kingdom
F.Ahmed@lboro.ac.uk

Stewart Robinson

School of Business and Economics
Loughborough University
Loughborough, LE11 3TU
United Kingdom
S.L.Robinson@lboro.ac.uk

Antuela Tako

School of Business and Economics
Loughborough University
Loughborough, LE11 3TU
United Kingdom
A.Takou@lboro.ac.uk

ABSTRACT

Conceptual modelling (CM) helps to determine the objectives, scope and content of a simulation model. It is widely agreed that CM is the most important phase in any simulation study. Despite its significance, it has only recently been recognized in the area of Modelling and Simulation (M&S). In Computer Science (CS), CM is also a pre-development phase for systems design when the requirements and objectives are being understood. However, within CS, there exist well-defined objectives, frameworks and quality evaluation methods for CM, and these have been in use for many decades. In this paper, we present a cross-disciplinary review of CM within the major fields of CS (Information Systems, Software Engineering and Databases) and make a comparison with M&S. The major contribution of this work is to highlight the need for a well defined process for CM within M&S along the lines of that in CS.

1 INTRODUCTION

Conceptual Modelling (CM) is widely used in Computer Science (CS) during the early phases of systems development to gather and communicate domain information for use in the design phase. Within Modelling and Simulation (M&S), a conceptual model helps to determine the objectives, inputs, outputs and content of the final simulation model (Robinson, 2008a). CM is regarded as the most important but least understood area of simulation study (Law, 1991).

Some well established standards, objectives and frameworks for CM now exist in the field of CS. However, within M&S, there are few processes for developing and testing a conceptual model. It has been proposed that rather than *reinventing the wheel* the integration of methods from these two fields would allow M&S to benefit from the CS methods.

There is little evidence of the adoption of CS methods by M&S despite the potential benefits. Various authors (Tolk and Turnitsa, 2012; Guizzardi and Wagner, 2012; Turnitsa et al, 2010; Tolk et al, 2013) have discussed the possibility of such integration; however, their research has mostly focused on typical CS theories, and some expertise of the subject matter is required to comprehend and put the ideas into practice. Also, these authors are more focused on establishing standard formalised definitions of M&S processes and do not provide any practical guidelines for beginners.

A more notable approach is presented by Arthur and Nance (2007), who investigated the possible use of Software Requirements Engineering (SRE) within the M&S discipline. Their paper discusses the potential to integrate the SRE lifecycle with the M&S lifecycle, but no guidance is given on how to integrate the lifecycles. Also, none of the above mentioned researchers make a detailed comparison of the similarities and differences between the fields of M&S and CS. This would help to identify the areas in both disciplines where any such integration would be appropriate.

The contribution of this paper is as follows: to perform a comparative study of CM in the fields of CS and M&S, thus highlighting the dearth of standards within M&S CM when compared with CS. This involves a comparison of four components of CM:

- Definition of CM
- Purpose of CM
- CM Process
- Conceptual Model Testing

The paper concludes by summarising the comparison of CM across the fields and outlining the next phase of our research. It also suggests an extension of this study around the approach of devising and comparing meta-models for CM in each domain.

2 CONCEPTUAL MODELLING IN COMPUTER SCIENCE (CS)

CM is probably the most discussed topic within CS, where it can be used for multiple purposes from an informal diagram in the early phases of a project to a very formal and machine readable document. In this section, we discuss CM within three major fields of CS, namely:

- Information Systems (IS)
- Software Engineering (SE)
- Databases (DB)

The following sub-sections are divided as per each of the above fields, and components of a CM, namely definition, purpose, process and testing, are discussed within each sub-section.

2.1 Information Systems (IS)

IS represents a major field within CS and deals with the management of information during system development. This can include the study of information flow for the design of IS in terms of hardware, software, people or enterprise. In this section, we present different perspectives on CM within IS.

2.1.1 IS: Definition of CM

According to Mylopoulos (1992), “conceptual modelling is the activity of formally describing some aspects of the physical and social world around us for the purpose of understanding and communication”. Siau (2004) concurs with this view, defining CM as a process of formally documenting a problem domain to further understanding and communication among stakeholders. Shanks et al (2003) describe CM as a representation (typically graphical) of an individual or group’s perception of a real world domain. This follows on from the definition of CM given by Wand and Weber (1990), who state that CM represents some aspects of the real world as perceived by humans. CM is agreed by most to be a formal process of describing and documenting a problem domain (physical and social world) to further understanding and communication. The scope of CM is the information about a real world domain.

2.1.2 IS: Purpose of CM

Wand and Weber (1990) define CM as an integral part of requirement analysis for IS development. They argue that CM should serve four purposes: a) improving communication between developers and users; b) helping analysts understand a domain; c) providing input for the design process; and d) documenting original requirements for future reference. According to Davies et al (2006), CM is also developed and used during the requirements analysis phase of IS development. Such methods are mostly graphic and are used to represent both static (e.g. entities) and dynamic (e.g. processes) phenomena in some domain. Other researchers agree that the purpose of CM is to promote understanding of domain requirements and to communicate these to developers and users. March and Allen (2009) claim that a conceptual modeller must analyse the domain to identify the important objects and rules, which must be understood in the context of system design and social constraints. Kung and Solvberg (1986) suggest that CM facilitates better communication between stakeholders, whilst Rolland and Cauvat (1992) describe CM as a bridge between user requirements and system design.

2.1.3 IS: Process of CM

CM processes in IS can broadly be assigned into two categories: theoretical (based on theories from the social and natural sciences, e.g. ontology, linguistics and cognition) and modelling (based on modelling languages, e.g. Unified Modelling Language (UML) and Entity-Relationship (ER) models). The theoretical aspect is based on creating meaningful representations from problem domains, whilst the modelling aspect is based on evaluating and refining modelling languages to map those representations onto suitable modelling constructs.

On the theoretical side, various researchers have proposed using ontology as a basis for a framework in CM. According to March and Allen (2009), the philosophical discipline of ontology provides a substantive basis for a shared understanding of CM. Evermann and Wand (2005) suggest using ontology to create meaningful definitions of the proposed system. They argue that modelling constructs should be based on the concepts of ontology since it can capture the behaviour, existence and relationships of objects in a more natural way than any other abstract theory. Wand et al (1995) present a theoretical framework based on human knowledge to support CM. They propose the use of ontology, concept theory and speech act theory (linguistics) as the main models of human knowledge to be used for the purpose of CM. On the modelling side, the most significant work is reported by Wand and Weber (2002), who present a set of guidelines/rules for modelling language, grammar and context. A similar framework is proposed by Guizzardi and Wagner (2005), but unlike Wand and Weber they do not provide explicit modelling rules.

In summary, support for CM processes within IS comes for the most part from a theoretical basis and modelling languages.

2.1.4 IS: Testing/Evaluation of Conceptual Model

Quality evaluation and testing of CM has also gained attention in the IS field. Some research has focused on evaluating the theoretical approaches used for domain presentation, whilst other work has aimed to either evaluate the general quality criteria or provide explicit guidelines for good quality CM.

Wand and Weber (1990) propose a framework based on Bunge’s ontology (Bunge, 1977). This is known as the Bunge-Wand-Weber (BWW) framework. BWW focuses on evaluation of the process quality involved in CM. An alternative framework based on linguistics was proposed by Lindland et al (1994), and this is known as LSS after Lindland, Sindre and Solvberg. LSS specifically focuses on evaluating the quality of the conceptual model itself. Both frameworks are discussed in some detail in section 2.3.4. Other work in this area has been carried out by Siau and Tan (2005), Evermann and Fang (2010) and Maes and Poels (2007). Parsons and Cole (2005) propose a set of guidelines for evaluating semantics during CM. These guidelines include criteria based on the choice of variables, procedures and participants. Gemino and Wand (2004) discuss a framework that shows the dependence between the modelling task, model creation and model interpretation. Also, Gemino and Wand (2005) examine the role of mandatory and optional properties in an ER-diagram for high quality CM. In summary, a quality framework for CM is mainly supported by a theoretical basis with some general quality guidelines also being available.

2.2 CM in Databases (DB)

Databases (DB) are primarily used to store and retrieve information within an organisation. This section covers the different components of CM within a DB system.

2.2.1 DB: Definition of CM

CM is considered to be a fundamental phase in DB design when knowledge about the UOD (Universe Of Discourse) is gathered and documented (Battista et al, 1989). An alternative view is that CM is a process by which the information content of a DB design is identified (Batra and Davis, 1992). In some studies, CM is considered to be the first formal specification of DB design (Engels et al, 1992).

2.2.2 DB: Purpose of CM

The CM of the DB produces a documented and structured form of knowledge describing some aspects of the UOD. Most of the research on DB CM focuses on creating meaningful representations of data in order to further understanding and enable sharing. According to Mineau et al (2000), the purpose of CM language is to capture the relevant knowledge (data) in an easy and understandable way. CM is used to capture a description of the phenomenon, its properties and their interactions for DB systems development (Battista et al, 1989). Crockett et al (1991) take a similar view, arguing that CM is an effective tool for managing an organisation's information and maintaining its integrity whilst at the same time enabling it to be shared amongst all members of the organisation. Promoting understanding and communication is considered to be the primary purpose of CM when used within DB systems.

2.2.3 DB: Process of CM

Most research on the DB CM process deals with the development or extension of modelling languages. The Entity-Relationship (ER) model, UML and conceptual graphs are mainly discussed. Some studies also support the combination of formal methods and ontological representation with the modelling languages. One of the earliest attempts made at DB CM was the ER model that was introduced by Chen (1976). An ER model is a combination of earlier proposed models with its own structure and presentation. This model, although widely accepted, has been criticised for a number of reasons, the main criticism being the lack of the expressiveness that is required to represent the complex inter-relationship between the objects and to capture the dynamic (behavioural) aspects of a DB design (Engels et al, 1992; Mylopoulos et al, 1990). Several extensions of the ER-model and its application to DB design have since been proposed (Engels et al, 1992; Mineau et al, 2000). Sugumaran and Storey (2002) recommend the use of ontology combined with the ER-model for CM in DB design. The method they propose includes mapping the ontological construct (entities, relationships, constraints) onto CM formalism (in this case they use the ER-model).

In summary, within DBs, modelling languages are mostly used for CM processes with some support from a theoretical basis.

2.2.4 DB: Testing/Evaluating of Conceptual Model

A number of important studies have focused explicitly on evaluating CM for DB design. Crockett et al (1991) propose a set of 12 criteria and sub-criteria to be used when developing CM for DBs. Their work is based upon the standards developed by ISO to improve the quality of a conceptual model. The four major characteristics identified by ISO for a DB CM include: ability to capture static and dynamic aspects; ability to communicate with both users and designers; easy and understandable; and adaptive to changes. Mineau et al (2000) highlight certain qualities that should be part of CM in DB design, whilst Sugumaran and Storey (2002) recommend using an ontology-based approach for the validation of CM in DB design. In order to evaluate quality in CM, they propose mapping quality criteria onto ontological constructs. A computer tool should then be used to check these mappings.

Various approaches are used to test the quality of a DB conceptual model, including ISO standards, the establishment of general guidelines and ontological support for the evaluation of CM.

2.3 CM in Software Engineering (SE)

This section discusses CM within Software Engineering (SE), which deals with the development of a software system as a sub-component of a large CS project or standalone application.

2.3.1 SE: Definition of CM

CM is regarded as an informal activity in SE that collects information about the real world domain from different stakeholders (users, analysts, developers). Kung (1989) defines CM as a model of the application domain. Similarly, Rolland and Prakash (2000) consider the CM process to be an analysis that provides information for users. CM has also been described as an abstraction from the real world domain (Andrade et al, 2004). Rolland and Prakash (2000) similarly describe CM as a process of abstracting the specifications of the required system. CM is also regarded as not only abstracting the specifications of real systems but also of proposed systems (systems to be built) (Kellner et al, 1999). Andrade et al (2004) define CM as a possible conceptual solution. Thus, a conceptual model is an abstraction of a real system and also of a proposed system. Both these views are discussed by Stavely (1983). In summary, for the most part, a conceptual model is considered to be an informal document that specifies the requirements of a real or proposed system.

2.3.2 SE: Purpose of CM

Within SE, CM is mainly believed to provide abstracted information from real world domains. Rolland and Prakash (2000) claim that the purpose of CM is abstraction from the problem domain, with the resulting conceptual model containing only relevant information about the system. According to Ares and Pazos (1998), CM is a very powerful abstraction technique. In SE, such abstraction significantly speeds up the analysis of an overly complex system (Ares and Pazos, 1998; Stavely, 1983). Furthermore, CM is an interactive process; once an abstracted model is generated, more and more constructs can be added to develop a more detailed model to be used as and when required (Kung, 1989). This model also helps to develop understanding of the system or problem since it provides a common reference platform upon which both the users and model developers can communicate (Ares and Pazos, 1998; Kung, 1989).

Thus, abstraction, understanding and communication are believed to be the most important objectives of CM in SE.

2.3.3 SE: Process of CM

CM is mostly used in the early stages of software development when the requirements are being gathered from the problem domain. Some CM frameworks are based on Software Requirements Engineering (SRE) methods, whilst others discuss the use of modelling languages or formal methods logic to create a consistent conceptual model.

An important framework based on SRE and object-oriented method is proposed by Insfrãn et al, 2002. Their CM is a UML based conceptual schema used to produce the final computer model. Rolland and Prakash (2000) discuss a framework which complements that of Insfrãn et al (2002), suggesting the use of SRE methods (particularly goal-oriented and scenario based methods) to produce a conceptual model. The framework proposed by Andrade et al (2004) recommends a viewpoint oriented approach for CM.

Kung (1989) proposes a framework that models the static aspects of the system using Entity-Relationship (ER) diagrams and the dynamic aspects of the system using traditional Data Flow Diagrams (DFDs). This method also includes mathematical logic to develop a conceptual model, which then undergoes a formal consistency check. Kung advocates combining these approaches in order to develop a verifiable and complete conceptual model. Delugach (1992) proposes the use of conceptual graphs (Sowa, 1984) to develop a common representation from four different language requirements, namely ER-diagram, DFD, state-diagram and requirements network.

In summary, in SE CM is mostly developed using SRE methods or modelling languages approaches. Some studies also support the use of formal methods logic.

2.3.4 SE: Testing/Evaluation of Conceptual Model

Most of the literature does not specifically focus on evaluating CM within the context of software development. However, there is still some support from a theoretical basis and the establishment of quality standards for CM within SE.

Some of the most important work on evaluating the quality of the CM process is presented by Moody (2005). He proposes the use of relevant international standards in software quality (ISO/IEC 9126) and quality management (ISO 9000) in evaluating a conceptual model. Dromey (1995) recommends the use of a similar framework to describe software quality models. Qi et al (2010) discuss the influence of CM on software reliability.

On the theoretical side, the most important framework is proposed by Lindland et al (1994). This framework is product oriented and aims to enhance the understanding of quality as it relates to CM. The essence of the framework lies in the distribution of quality across syntactic (structure), semantic (meaning) and pragmatic (interpretation by audience) categories. These important linguistic concepts are applied to four aspects of modelling in the framework, namely language, domain, model and audience participation. The framework of Wand and Weber (1990) discussed in 2.1.4 is also used in SE to evaluate the quality of CM from a process oriented view. Nelson and Monarchi (2007) discuss a model that combines approaches from both these frameworks to evaluate the quality of CM.

ISO standards are mostly used for evaluating CM quality. Some useful support is available for the evaluation of general quality criteria along with a theoretical basis for testing a conceptual model.

3 CM IN M&S

Until recently, little work has been done in the field of M&S to promote the importance of CM. CM has been recognised as an integral part of M&S development as the complexity and size of M&S applications have continued to increase (Arthur and Nance, 2007; Balci and Ormsby, 2007; Robinson, 2002; Robinson, 2004; Robinson, 2008a). In this section, CM within M&S is discussed using the same four components as for the discussion above.

3.1 M&S: Definition of CM

CM is not well defined in the simulation literature. However, it is generally accepted that CM is part of the early stage of a simulation study (Robinson, 2008a).

Robinson (2008a) defines a conceptual model as “a non-software specific description of the computer simulation model (that will be, is or has been developed), describing the objectives, inputs, outputs, content, assumptions and simplifications of the model.” According to this view, a conceptual model is the simplified representation of a simulation model that may or may not exist. This view is in contrast to that of Balci et al (2008), who state that “A simulation conceptual model is a repository of high level conceptual constructs and knowledge specified in a variety of communicative forms.” According to Landry et al (1983), a conceptual model is the mental image of the problem situation formed by the perceptions of the modeller and decision-makers. This mental image may just be present in the minds of the modellers or users.

Most studies support informal ways of documenting a conceptual model with some support for formal documentation being present. The scope of CM covers models of real or proposed systems, while some regard it as a mental model of the problem domain.

3.2 M&S: Purpose of CM

According to Kotiadis and Robinson (2008), CM is a process by which a model is abstracted from the real world. Adding to this, Robinson (2011) declares that CM is a process of deciding what to model. This view of abstraction is supported by Tolk and Turnitsa (2012), who state that modelling is a useful process of abstracting theories from a system and capturing the elicited information in a conceptual model. CM helps to determine the modelling objectives along with the general project objectives (Robinson, 2008b). According to Guizzardi and Wagner (2012), the main purpose of CM is to capture a part of the real-world domain under consideration. A conceptual model may be written in a form that is accessible to different stakeholders (managers, analysts and model developers) (Balci and Ormsby, 2007). According to Tolk and Turnitsa (2012), a model is a design guidance and inspiration for simulation. This perception is supported by Robinson (2008a).

In summary, the purpose of CM is reported as being to further understanding and conduct analysis of the problem domain, whilst some cite abstraction as a goal and others regard CM as a means of communication. These are related, but quite different motivations.

3.3 M&S: Process of CM

There are few well defined frameworks in the area of CM despite it being recognised as the most important phase in the simulation project process (Robinson, 2008a). The existing frameworks are mostly descriptive in nature, with some support from modelling languages.

One of the most well accepted descriptive frameworks was proposed by Robinson (2008b). The two major phases of CM as described by Robinson (2011) are knowledge acquisition and model abstraction. While the first phase helps to form the initial system description, the second phase generates a possible conceptual model of the problem. Also, according to Robinson (2004), the CM process involves the following activities: 1) Developing an understanding of the problem situation; 2) Determining the modelling objectives; 3) Designing the CM: inputs, outputs and model content; and 4) Collecting and analysing the data required to develop the CM. Some other descriptive and diagrammatic frameworks are discussed in Robinson et al (2011). An extension to Robinson's (2008b) framework has recently been proposed by Chwif et al (2013), who have added complexity description, scenario description and input/output description within the same block.

Kotiadis (2007) proposes the use of Soft System Methodology (SSM) in the CM objectives phase, and Montevechi and Friend (2012) suggest using a five step framework based on SSM to achieve the phases of a CM defined in Robinson (2008b). The use of SSM has been acknowledged in the area of participative and facilitated CM (Tako et al, 2010; Tako and Kotiadis, 2012). An interesting aspect of the work in this area is the development of the PartiSim (Participative Simulation Modelling) framework by Tako et al (2010).

van der Zee and Van der Vorst (2007) present diagrammatic techniques to support CM creation in manufacturing simulation, whilst Balci and Ormsby (2007) propose a detailed simulation CM lifecycle, which includes problem formulation, system definition, high-level CM, detailed design and CM specification

Some proposals for adapting the theoretical basis and techniques for CM have recently emerged in the Winter Simulation Conferences. Turnitsa et al (2010) suggest the use of ontology when forming the research question that exists in the mind of the modeller, while Guizzardi and Wagner (2012) propose the use of UML class diagrams and Business Process Modelling Notation with some improved ontological foundation. Interesting work in this area has been conducted by Tolk and Turnitsa (2012), who propose a process driven approach towards CM. The problem with these proposals (as mentioned in the introduction section) is that they require expert knowledge of the subject matter to begin with and they are more focused on formalisation of the processes rather than on describing a structured method for novices to use.

In summary, a cluster of scattered frameworks for use in CM exist. From these, the descriptive frameworks have provided a good foundation for a CM framework; however, there is an acute need for guided methods which can provide step by step instructions for the development of a structured conceptual model.

3.4 M&S: Testing/Evaluation of Conceptual Model

Unlike in the field of CS, no theoretical frameworks exist in M&S to test the quality criteria of a conceptual model (for example, ontology based or cognition based). Work on deciding requirements, criteria and Verification and Validation (V&V) of simulation models in operational research is ongoing (Kleijnen, 1995; Robinson, 1997; Sargent, 2011; Oral and Kettani, 1993; Landry et al, 1983; Balci and Ormsby, 2007; Sargent, 2005). However, the focus is not on the quality of CM and discussion is domain specific (some are only relevant for large scale designs).

The four key requirements of a conceptual model as presented in Robinson (2008a) are validity, credibility, utility and feasibility. These criteria are the foundations on which a conceptual model should be developed to ensure that it is possible to address the modelling objectives within the time and resources available. Sargent (2005) discusses conceptual model validity in a more scientific manner. This is because he views CM as a mathematical, verbal and logical representation of the system under consideration. The simulation CM life cycle presented by Balci and Ormsby (2007) contains various methods for verification and validation at each stage of a simulation study.

In summary, for the most part, the literature discusses general quality criteria, but there is a notable absence of any well formed frameworks built upon strong theoretical foundations (such as ontology and linguistics). Some verification and validation lifecycles are discussed, but these remain domain specific.

4 CONCLUSIONS

The components of CM (definition, purpose, process, testing) can be assigned the following characteristics within all fields (IS, DB, SE and M&S):

- *Definition of CM*: level of formality, scope
- *Purpose of CM*: understanding, communication, abstraction
- *CM Process*: theoretical basis, formal methods, modelling languages
- *Conceptual Model Testing*: quality standards and criteria

Our discussion on these components and characteristics is summarised in table 4.1., and the contents of the table can be used to draw conclusions. For example, we can conclude from information in the process row and within the SE column that the majority of the processes for CM are based on modelling languages, followed by some structured SRE methods, with very little support from formal methods or mathematical formalism. Similarly, other grids of the table can be interpreted and compared with each other for quick analysis of the literature on each aspect of CM between CS (IS,DB,SE) and M&S.

A closer look at table 4.1 reveals some interesting features with respect to our area of interest, M&S. M&S definitions of CM are mostly informal and can include real or proposed systems. This perspective is closest to the SE perspective of CM. The objectives of CM within M&S are mostly reported to be related to abstraction (hiding irrelevant details when transforming information for analysis purposes). This view is again close to the SE view of abstraction of CM. Processes for CM within M&S are mostly descriptive in nature, with little work based on modelling languages. This is in contrast to all other fields of CS, where we see mostly structured/guided methods with some good theoretical basis. Similarly, CM testing within M&S is lacking, there being few well formed methods for evaluating quality. These findings could be extended by using a framework based on meta-models (upper ontologies) for each CM domain. This would enable a broad comparison of the different fields so that more components of CM (relations, types, attributes, properties etc.) across all domains can be compared. Such an approach could bring greater precision in identifying the potential areas of improvements for CM in M&S.

This discussion leads us in a clear future direction. Some urgent work is required in the process and testing areas of CM within M&S along the lines of CS so as to gain the same benefits as are currently available in the latter field (mainly consistency, formality, communication and evaluation).
Such work has already begun as a part of ongoing research on SRE methods based on view point oriented requirements techniques (VORD, PRE-view) (Kotonya and Sommerville, 1996; Sommerville et al, 1998) are being adapted for the CM process with the necessary modifications being made to suit the simulation modelling context.

Table 4.1 *Comparison of different components in CM across CS and M&S*

Components of CM	IS	DB	SE	MS
Definition	Documented through formal means and scope is mostly model of a real world	Expressed as a formal document and scope is to cover the model of real world information	Expressed as informal diagrams at the start of system development process. Scope can cover model of real or proposed system	Generally informal ways of documentation and rarely requires a formal document. Scope is a model of real or proposed system, but some view a conceptual model as a mental model of how to simulate the problem
Objectives	he purpose is for understanding the problem domain and presenting it in a communicative form	The main purpose is to be representation of information in a form which is understandable to the designer of DB	Used for understanding initial requirements for the existing or proposed system and communicating this to the design phase. Abstraction is used to hide irrelevant requirements and simplify the prototype for the design phase	Understanding the problem domain and presenting it through abstraction to the model domain. Abstraction here serves to transfer the initial system description to a conceptual model. Communicative forms of conceptual model can also help with model design.
Process	Most major frameworks for CM have a theoretical basis (ontology/linguistics).Some also discuss the use of modelling languages e.g. UML and ER models	Focus on using modelling languages that include UML and ER models. Few studies have reported the use of formal methods e.g. temporal logic.	Methods are mostly based on modelling languages/ graphical method (conceptual graphs/ER models/data flow diagrams). Some important work is based on SRE methods (view-point oriented, extreme programming etc). There are also formal methods approaches using mathematical logics	Most frameworks are descriptive and informal e.g. using tables, flow charts and text. Some are based on modelling languages (UML, SysML or SAD). There is a lack of guided frameworks from established methods and formal techniques
Testing/Evaluation	Most frameworks for quality testing are based on theory (ontology/linguistics). Some of them are concerned with general quality criteria and guidelines	A few important frameworks have established ISO standards. Some are formed on a basis of theoretical approach (ontology/linguistics). Others focus on general criteria/guidelines	ISO standards based framework is reported along with some general qualities or characteristics evaluation. Some work supports the theoretical basis (ontology) for quality evaluation.	Lack of any well-formed frameworks/methods based on theory. Mostly general quality criteria are discussed. Some use verification life-cycles, but these are domain specific.

REFERENCES

Andrade J, Ares J, Garcia R, Pazos J, Rodriguez S and Silva A (2004). A methodological framework for viewpoint-oriented conceptual modeling. *IEEE Transactions on Software Engineering.* 30: 282-294.

Ares J and Pazos J (1998). Conceptual modelling: an essential pillar for quality software development. *Knowledge-Based Systems.* 11: 87-104.

Arthur J D and Nance R E (2007). Investigating the use of software requirements engineering techniques in simulation modelling. *Journal of Simulation.* 1: 159-174.

Balci O, Arthur J D and Nance R E (2008) Accomplishing reuse with a simulation conceptual model. *Proceedings of the 40th Winter Simulation Conference*: Miami, Florida.

Balci O and Ormsby W F (2007). Conceptual modelling for designing large-scale simulations. *Journal of Simulation.* 1: 175-186.

Batra D and Davis J G (1992). Conceptual data modelling in database design: similarities and differences between expert and novice designers. *International Journal of Man-Machine Studies.* 37: 83-101.

Battista G D, Kangassalo H and Tamassia R (1989). Definition libraries for conceptual modelling. *Data & Knowledge Engineering.* 4: 245-260.

Bunge M (1977). Ontology I: The Furniture of the World (vol. 3): Reidel publishers, Dordrecht, Holland.

Chen P P-S (1976). The entity-relationship model: towards a unified view of data. *ACM Transactions on Database Systems*. 1: 9-36.

Chwif L, Banks J, de Moura Filho J P and Santini B (2013). A framework for specifying a discrete-event simulation conceptual model. *Journal of Simulation.* 7: 50-60.

Crockett H D, Guynes J and Slinkman C W (1991). Framework for development of conceptual data modelling techniques. *Information and Software Technology*. 33: 134-142.

Davies I, Green P, Rosemann M, Indulska M and Gallo S (2006). How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering.* 58: 358-380.

Delugach H S (1992). Specifying multiple-viewed software requirements with conceptual graphs. *Journal of Systems and Software.* 19: 207-224.

Dromey R G (1995). A model for software product quality. *IEEE Transactions on Software Engineering.* 21: 146-162.

Engels G, Gogolla M, Hohenstein U, Hulsmann K, Lohr-Richter P, Saake G and Ehrich H-D (1992). Conceptual modelling of database applications using an extended ER model. *Data & Knowledge Engineering.* 9: 157-204.

Evermann J and Fang J (2010). Evaluating ontologies: Towards a cognitive measure of quality. *Information Systems.* 35: 391-403.

Evermann J and Wand Y (2005). Ontology based object-oriented domain modelling: fundamental concepts. *Requirements Engineering.* 10: 146-160.

Gemino A and Wand Y (2004). A framework for empirical evaluation of conceptual modeling techniques. *Requirements Engineering.* 9: 248-260.

Gemino A and Wand Y (2005). Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties. *Data & Knowledge Engineering.* 55: 301-326.

Guizzardi G and Wagner G (2005). Towards ontological foundations for agent modelling concepts using the unified fundational ontology (UFO). *Agent-Oriented Information Systems Ii.* 3508: 110-124.

Guizzardi G and Wagner G (2012) Conceptual simulation modeling with onto-UML. *Proceedings of 2012 Winter Simulation Conference* : Berlin, Germany

Insfran E, Pastor O and Wieringa R (2002). Requirements Engineering-Based Conceptual Modelling. *Requirements Engineering.* 7: 61-72.

Kellner M I, Madachy R J and Raffo D M (1999). Software process simulation modeling: Why? What? How? *Journal of Systems and Software.* 46: 91-105.

Kleijnen J P C (1995). Verification and validation of simulation models. *European Journal of Operational Research.* 82: 145-162.

Kotiadis K (2007). Using soft systems methodology to determine the simulation study objectives. *Journal of Simulation.* 1: 215-222.

Kotiadis K and Robinson S (2008) Conceptual modelling: knowledge acquisition and model abstraction. *Proceedings of the 40th Winter Simulation Conference*: Miami, Florida

Kotonya G and Sommerville I (1996). Requirements engineering with viewpoints. *Software Engineering Journal.* 11: 5-18.

Kung C H (1989). Conceptual Modeling in the Context of Development: *IEEE Transactions on Software Engineering.* 15: 1176.

Kung C H and Solvberg A (1986). Activity modeling and behavior modeling. *In: Proceedings of the IFIP WG 8.1 working conference on Information systems design methodologies*: improving the practice: Amsterdam, The Netherlands: North-Holland Publishing Co, 145-171.

Landry M, Malouin J-L and Oral M (1983). Model validation in operations research. *European Journal of Operational Research.* 14: 207-220.

Law A M (1991). Simulation model's level of detail determines effectiveness. *Industrial Engineering.* 23: 16-18.

Lindland O I, Sindre G and Solvberg A (1994). UNDERSTANDING QUALITY IN CONCEPTUAL MODELING. *IEEE Software.* 11: 42-49.

Maes A and Poels G (2007). Evaluating quality of conceptual modelling scripts based on user perceptions. *25th International Conference on Conceptual Modeling (ER 2006) Four of the best papers presented Some issues in privacy data management (ICDE 2006).* 63: 701-724.

March S T and Allen G N (2009). Challenges in Requirements Engineering: A Research Agenda for Conceptual Modeling. *Design Requirements Engineering: A Ten-Year Perspective.* 14: 157-165.

Mineau G W, Missaoui R and Godinx R (2000). Conceptual modeling for data and knowledge management. *Data & Knowledge Engineering.* 33: 137-168.

Montevechi J A B and Friend J D (2012) Using a soft systems methodology framework to guide the conceptual modeling process in discrete event simulation. *Proceedings of the 2012 Winter Simulation Conference.* Berlin, Germany.

Moody D L (2005). Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering.* 55: 243-276.

Mylopoulos J (1992). Conceptual Modeling and Telos. *In:* Loucopoulos P andZicari R (eds.) *Conceptual modeling, databases, and case : an integrated view of information systems development:*Wiley publishers , Newyork.

Mylopoulos J, Borgida A, Jarke M and Koubarakis M (1990). Telos: representing knowledge about information systems. *ACM Transaction on .Information Systems.* 8: 325-362.

Nelson H and Monarchi D (2007). Ensuring the quality of conceptual representations. *Software Quality Journal.* 15: 213-233.

Oral M and Kettani O (1993). The facets of the modeling and validation process in operations research. *European Journal of Operational Research.* 66: 216-234.

Parsons J and Cole L (2005). What do the pictures mean? Guidelines for experimental evaluation of representation fidelity in diagrammatical conceptual modeling techniques. *Data & Knowledge Engineering.* 55: 327-342.

Qi Y-d, Xie X-f, Zhu A-h and Yan X-b (2010). Analysis of contribution of conceptual model quality to software reliability. *In: Computer Application and System Modeling (ICCASM) 2010*

Robinson S (1997) Simulation model verification and validation: increasing the users' confidence. *Proceedings of the 29th Winter Simulation conference:* Atlanta, Georgia, USA IEEE Computer Society.

Robinson S (2002). Modes of simulation practice: approaches to business and military simulation. *Simulation Modelling Practice and Theory.* 10: 513-523.

Robinson S (2004). Discrete-event simulation: from the pioneers to the present, what next? *Journal Of Operational Reserach Society*. 56: 619-629.
Robinson S (2008a). Conceptual modelling for simulation Part I: definition and requirements. *Journal of the Operational Research Society*. 59: 278-290.
Robinson S (2008b). Conceptual modelling for simulation Part II: a framework for conceptual modelling. *Journal of the Operational Research Society* . 59: 291-304.
Robinson S (2011). Choosing the right model: Conceptual modeling for simulation. *In: Proceedings of 2011 Winter Simulation Conference*: 1423-1435.
Robinson S, Brooks R, Kotiadis K and Van der zee D-J (2011) *Conceptual modeling for Discrete Event Simulation*: Boca Raton London Newyork, CRC Press.
Rolland C and Cauvat C (1992) Trends and Perspectives in Conceptual Modelling. *Universitaƒa Panthaƒade la Sorbonne (Paris 1).*
Rolland C and Prakash N (2000). From conceptual modelling to requirements engineering. *Annals of Software Engineering.* 10: 151-176.
Sargent R G (2005) Verification and validation of simulation models. *Proceedings of the 37th winter Simulation Conference*: Orlando, Florida
Sargent R G (2011). Verification and validation of simulation models. *In:* Proceedings of the 2011 Winter Simulation Conference (WSC): 183-198.
Shanks G, Tansley E and Weber R (2003). Using ontology to validate conceptual models. *Communications of the Acm.* 46: 85-89.
Siau K (2004). Informational and computational equivalence in comparing information modeling methods. *Journal of Database Management (JDM).* 15: 73-86.
Siau K and Tan X (2005). Improving the quality of conceptual modeling using cognitive mapping techniques. *Quality in conceptual modeling Five examples of the state of art The International Workshop on Conceptual Modeling Quality 2002 and 2003.* 55: 343-365.
Sommerville I, Sawyer P and Viller S (1998) Viewpoints for Requirements Elicitation: A Practical Approach. *Proceedings of the 3rd International Conference on Requirements Engineering: Putting Requirements Engineering to Practice.* IEEE Computer Society.
Sowa J F (1984) *Conceptual structures : Information processing in mind and machin*: USA, Addison-Wesely Publ. Co, Reading, Mass.
Stavely A M (1983). Modeling and projection in software development. *Journal of Systems and Software.* 3: 137-146.
Sugumaran V and Storey V C (2002). Ontologies for conceptual modeling: their creation, use, and management. *Data & Knowledge Engineering.* 42: 251-271.
Tako A A and Kotiadis K (2012) Facilitated conceptual modelling: practical issues and reflections. *Proceedings of 2012 Winter Simulation Conference*: Berlin, Germany.
Tako A A, Vasilakis C and Kotiadis K (2010). A participative modelling framework for developing conceptual models in healthcare simulation studies. *In: Proceedings of the 2010 Winter Simulation Conference (WSC)*: 500-512.
Tolk A, Diallo S Y, Padilla J J and Herencia-Zapana H (2013). Reference modelling in support of M&S: foundations and applications. *Journal of Simulation.* 7: 69-82.
Tolk A and Turnitsa C (2012). CONCEPTUAL MODELING WITH PROCESSES. *In: Proceedings of 2012 Winter Simulation Conference*: US.
Turnitsa C, Padilla J J and Tolk A (2010). Ontology for Modeling and Simulation. *In:* Proceedings of 2010 Winter Simulation Conference : 643-651.
van der Zee D J and Van der Vorst J G A J (2007). Guiding principles for conceptual model creation in manufacturing simulation. *In: Proceedings of 2007 Winter Simulation Conference*: 776-784.
Wand Y, Monarchi D E, Parsons J and Woo C C (1995). Theoretical foundations for conceptual modelling in information systems : *Decision Support Systems.* 15: 285-304.
Wand Y and Weber R (1990). An ontological model of an information system: *IEEE Transactions on Software Engineering.* 16: 1282-1292.

Wand Y and Weber R (2002). Research commentary: Information systems and conceptual modeling: A research agenda. *Information Systems Research.* 13: 363-376.

AUTHOR BIOGRAPHIES

FAHIM AHMED is a PhD student at the School of Business and Economics, Loughborough University. He holds an MSC in Control Engineering from Chalmers University of Technology, Sweden. Previously, he has worked with the project that involves formal verification of Discrete Event Systems at the department of computing and software, McMaster University, Canada. Key areas of interest are simulation modelling for discrete event systems, application of software engineering methods to simulation modelling and cross-disciplinary study between computer science and operational research.

ANTUELA A. TAKO is a Lecturer in Operations Research at the School of Business and Economics, Loughborough University. She holds a PhD in Simulation and an MSc in Management Science and Operational Research from the University of Warwick. She previously worked for a research project that introduces stakeholder participation and facilitation in discrete-event simulation modelling. Her research interests include the comparison of simulation approaches (discrete-event simulation and system dynamics), participative simulation modelling and conceptual modelling. Home page:
http://info.lut.ac.uk/departments/sbe/staff/academic-research/bsat2.html

STEWART ROBINSON is Professor of Management Science and Associate Dean Research at Loughborough University, School of Business and Economics. Previously employed in simulation consultancy, he supported the use of simulation in companies throughout Europe and the rest of the world. He is author/co-author of five books on simulation. His research focuses on the practice of simulation model development and use. Key areas of interest are conceptual modelling, model validation, output analysis and alternative simulation methods (discrete-event, system dynamics and agent based). Professor Robinson is co-founder of the Journal of Simulation and President of the Operational Research Society. Home page: www.stewartrobinson.co.uk.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

SIMULATION OF COMPETITION IN REVENUE MANAGEMENT

Dr Christine S.M. Currie

Mathematics
University of Southampton
Highfield, Southampton
SO17 1BJ
christine.currie@soton.ac.uk

ABSTRACT

This paper describes the use of simulation to help solve a revenue management problem. In revenue management we aim to maximise the revenue from a fixed set of products either by optimising the prices to be charged for each of the products or by an optimal allocation of inventory to fixed prices. In this project, we consider a competitive market where potential customers use price as their main decision variable. The simulation model is then used to find some key characteristics of the revenue management problem in competitive situations. There has been some research into simulation in revenue management and a brief overview of the literature will also be given.

Keywords: Revenue management, competition, simulation

1 INTRODUCTION

The revenue management (RM) problem is essentially that of maximizing the revenue made from a stock of products. Arguably the most popular definition was put forward by Robert Cross, CEO of Revenue Analytics: “the objective of Revenue Management is to sell the right product to the right customer at the right time for the right price”. In this work, as is true of much of the academic work in RM, we consider the problem of maximizing the revenue made from a fixed stock of identical, perishable products or services. By perishable we mean that beyond a certain date the products are worthless. The most common application of this problem is the sale of airline tickets and this is certainly where RM is most widely used. However, it is also applicable to the sale of cabins on cruise liners, cinema tickets, coach and rail tickets, and in the sale of fashion or seasonal goods in retail.

The particular problem we focus on here is the optimization of prices, termed optimal or dynamic pricing. Early research into RM tended to concentrate on quantity-based RM, whereby a set of prices or fares are determined at the start of the booking period and the organization offering the product or service must optimize the number available for purchase at each of the fare levels (e.g. Littlewood, 1972 and Belobaba 1989). In contrast, when using price-based RM the aim is to set prices throughout the booking period such that all of the products are sold to the people who place the highest valuation on them. Gallego and van Ryzin (1997) argue that optimal pricing is the preferred option as it reduces sales by increasing price rather than by limiting supply, and this will necessarily lead to higher revenues.

The simulation model that we describe in this paper follows on from work carried out with a major UK airline to determine the optimal pricing policy for tickets under “one-way” pricing, where tickets are bought on a leg by leg basis rather than as paired returns or multi-leg trips. Under this policy, each ticket sold corresponds to an identical product with the same rules and restrictions, and the ticket price is only determined by the time left before the flight leaves (Anjos et al., 2004 and 2005; Currie et al., 2008). In this paper, we introduce the effect of price competition, assuming that potential customers will choose whether to fly and, if so, which company to fly with, based on the fares that are available.

In the next section we give a brief overview of the optimal pricing literature, particularly focusing on how it applies to competition in RM. We also discuss the use of simulation in previous RM studies. Following on from that, we describe the simulation model of price competition in Section 3, give some preliminary results in Section 4 and discuss the conclusions and possible future directions for the work in Section 5.

2 LITERATURE REVIEW

The first research into optimal pricing was carried out in the 1960s (Kincaid and Darling, 1963), but it has only taken off in the past decade, mainly due to the increased importance of the Internet for selling goods and services. It is now much easier for customers to compare the prices offered by different companies and also much cheaper for companies to change the prices being charged during the selling period. The basic optimal pricing model assumes that customers arrive at a store (or access a website) at a given rate and will then buy a unit of the product or service on offer with a given probability. The expected number of sales is then equal to the number of potential customers multiplied by the probability that they will buy the product, summed over the selling period.

As discussed in the Introduction, we worked with a UK airline to develop a method for optimizing prices under one-way pricing. The initial work concentrated on finding the theoretical properties of optimal pricing functions and developing a practical methodology for parameterizing the models (Anjos et al., 2004 and 2005). Lagrangian multipliers and calculus of variations were used to find the optimal pricing function and we were able to prove that under the assumptions we made about customer behavior, the optimal price of tickets should always be increasing, justifying a common practice in airline RM. However, competition has long been acknowledged as an important missing element in pricing models (Bitran and Caldentey, 2003; Elmaghraby and Keskinocak, 2003; Zhao and Zheng, 2000) and was believed by the airline to be a key influence on customer behavior. Consequently, we went on to consider a very simple example of competition – a duopoly with both companies offering identical products. This yielded some interesting theoretical results but to be more realistic needs to be able to take account of competition between more than two companies. The simulation model that we describe here uses the same ideas as this earlier work but extends it to a market with many players.

Other recent work in the optimal pricing literature which considers competition between different products makes use of customer choice behavior (e.g. Talluri and van Ryzin, 2004; van Ryzin and Vulcano, 2008). Potential customers are assumed to arrive at the seller following an inhomogeneous Poisson process and will purchase the product with a probability dependent on its price and other characteristics of the product. The multinomial logit model (MNL), described by Simon Anderson et al. (1992) and by Ben-Akiva and Lerman (1985) is widely used to describe the probability of purchase. The majority of papers consider the easier problem of competition between different products offered by the same seller (Talluri and van Ryzin, 2004; van Ryzin and Vulcano, 2008a; Vulcano et al., 2010) rather than competition between two independent companies.

Simulation is principally used in RM for testing of algorithms (e.g. Graf and Kimms, 2011; Gorin and Belobaba, 2008), although ideas from simulation optimization; in particular the stochastic approximation algorithm developed by Robbins and Munro (1951), have been widely used to improve heuristics for the solution of a particular kind of optimization problem (e.g. van Ryzin and Vulcano, 2008b; Kunnumkal and Topaloglu 2010; Levina et al., 2011). The model that comes closest to what we are trying to do here is PODS, a simulation model developed in the late 1990s by Boeing and MIT to be a tool for simulating a competitive market, where airlines compete over a network of origin-destination markets (Lee, 1998). PODS consists of four sub-models that describe 1. Customer choice; 2. Airlines’ RM strategies; 3. Forecasting of key parameters based on simulated outputs; 4. Historical database. Potential customers arrive and make a purchase decision based on a disutility model. The airlines in the simulation will then make a decision about whether to accept the purchase given the RM strategy that they are employing. Belobaba and Wilson (1997) describe how PODS can be used to answer key questions in RM as well as providing more details of the model itself.

3 SIMULATION MODEL

3.1 Arrival Process

We assume that the arrival of customers follows a non-homogeneous Poisson process with a rate parameter $f(t)$ that is dependent on the time left until the end of the booking period,

$$f(t) = (f + dt)\exp(-ht). \qquad (1)$$

The assumption of a non-homogeneous Poisson process is common in optimal pricing projects (e.g. Zhao and Zheng, 2000; Talluri and van Ryzin, 2005)and the form we use is justified by observations of real booking data, where we tend to observe few bookings early on with an increase to a peak rate of bookings close to departure.

3.2 Customer Choice

Potential customers make a decision over whether to purchase a ticket and which company to buy from using a probability function that depends on the time left until the end of the booking period t, and the vector of prices on offer from all of the companies in the market. In the future we aim to consider different scenarios for how customers choose their tickets, e.g. whether they take account of the minimum or maximum price in the market or compare each of the prices with the average of the prices on offer. In this paper, we just consider the situation where they compare with the minimum price. The probability that a customer will purchase a product from company i is therefore written as

$$p_i(\boldsymbol{y}, t) = \frac{1}{n} exp\left[-y_i(t)(a + bt) - \frac{y_i(t)}{Min\{y_i(t)\}}(p_i + q_i t)\right], \qquad (2)$$

where there are n companies competing in the market and $y_i(t)$ is the price being charged by company i at time t; Min$\{y_i(t)\}$ is equal to the minimum of the prices offered. The form of the probability function is similar to that used in previous work on RM in airlines (Anjos et al., 2004 and 2005). This particular form is chosen based on the assumption that a customer will select a product with a probability dependent on its price and the lowest available price in the market place. The probability of purchase of product i decreases with increasing price and when the price of product i is high in comparison with the minimum price on offer. We also include a time dependence such that the probability of purchase for a given set of prices increases as we approach departure.

We allow the parameters p_i and q_i to vary between the different companies to express the fact that customers may have some underlying preference for one of the companies in the market.

3.3 Revenue Management

In this version of the simulation we do not assume that the companies react to other prices in the market or to the number of bookings that they have received. Instead, prices are set at the start of the selling period. The model is set up so that we could incorporate price updates and this would certainly make the simulation more realistic.

3.4 Simulation Model

The model is written in Visual Basic, attached to an Excel spreadsheet. The structure of the model is shown in Figure 1.

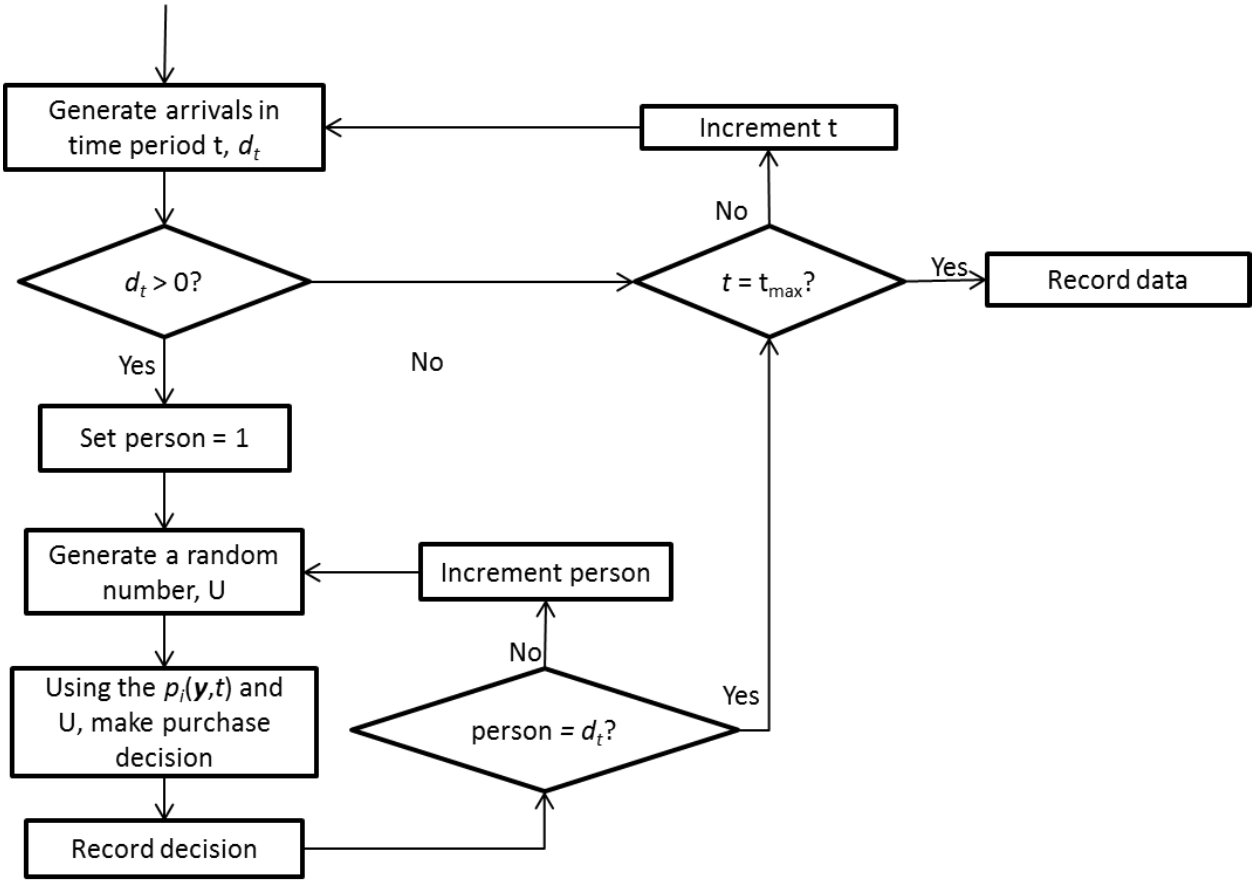


Figure 1 *Structure of the RM Simulation model*

4 EXAMPLE

We consider an example of a three-player market where player 1 is the most desirable company and player 3 is the least desirable. The price structures are set up to reflect this with player 1 charging the highest prices and player 3 charging the lowest prices. We set p_i and q_i to reflect the differences in customer preference between the companies. If all 3 companies are charging the same fare, customers would purchase from player 1 with the highest probability. Such markets are not uncommon, e.g. consider a flight route served by British Airways, bmi and RyanAir. The majority of customers would choose to fly with British Airways if the price being offered were the same as that offered on RyanAir.

The parameter values used in the simulation are given in Table 1. The parameter values result in a situation where many people browse the prices early on in the booking period but the probability of any of them making a purchase is very low. In the last few days of the booking period, nearly all of the people looking at flight prices will make a purchase with one of the companies.

Table 1 *Parameter Values Used in the Simulation Model*

Parameter	Company 1	Company 2	Company 3
p	0.03	0.04	0.05
q	0.03	0.04	0.05
a	0.0001		
b	0.00005		
f	60		
d	32		
h	0.125		

We run the model for 100 iterations to obtain the expected revenue and booking numbers for each of the companies. Of particular interest to us is the strategy that should be employed by company 2 in order to maximise its revenue. Company 2 is the middle player, less preferred to company 1 but not able to offer such low prices as company 3. In the scenario considered here, we examine the situation where company 2 uses the strategy of pricing itself £10 below company 1 in each time period, and ignores the impact of company 3. Company 3 is assumed to aim to offer the cheapest price at all times, while company 1 is the market leader. The prices being charged are shown in Figure 2.

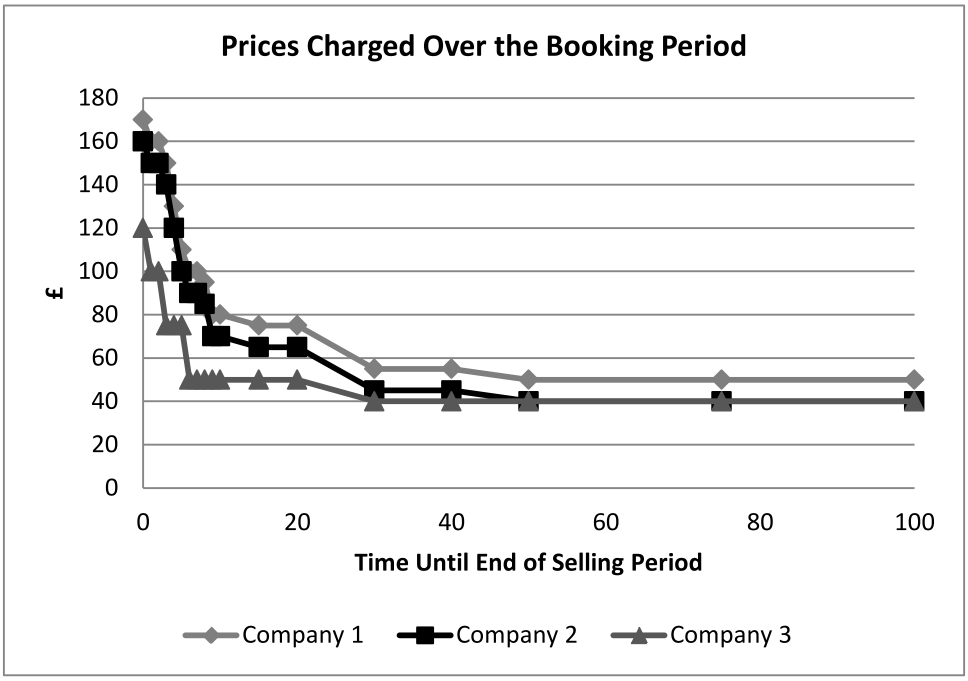


Figure 2 *Prices being Charged by the Companies During the Booking Period*

We can see from the revenue results presented in Figure 3 that company 1 gains the most revenue because of its position in the market as the preferred player. This allows it to charge significantly more during the final few days of the booking period and still draw in a significant number of passengers, as demonstrated in Figures 4 and 5.

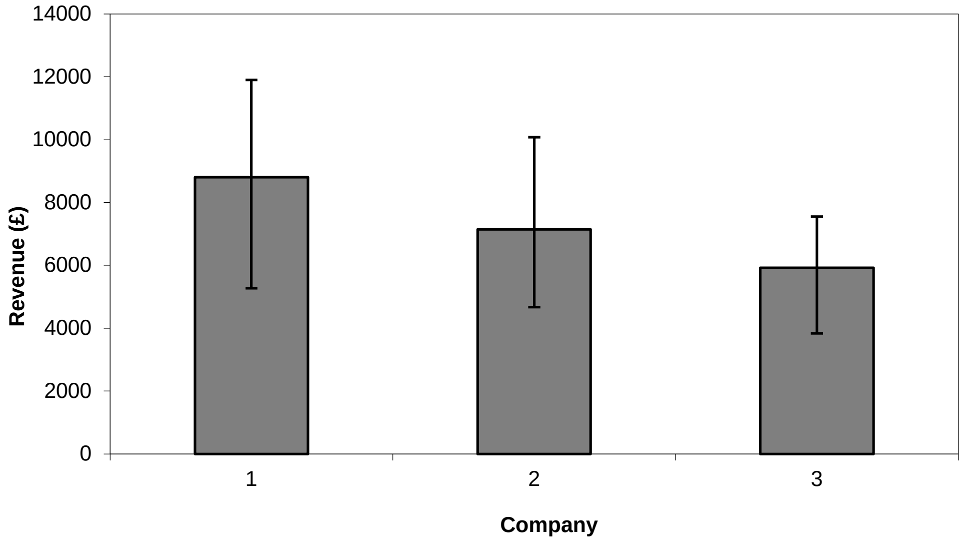


Figure 3 *Expected Total Revenue for the Three Companies*

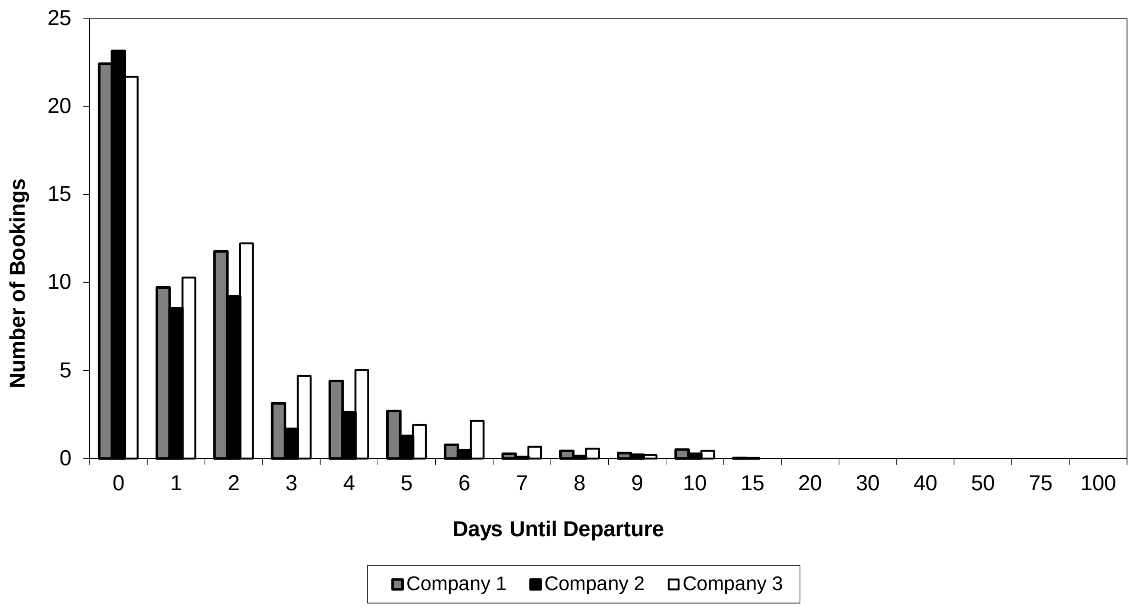


Figure 4 *Expected Numbers of Bookings between Reading Days*

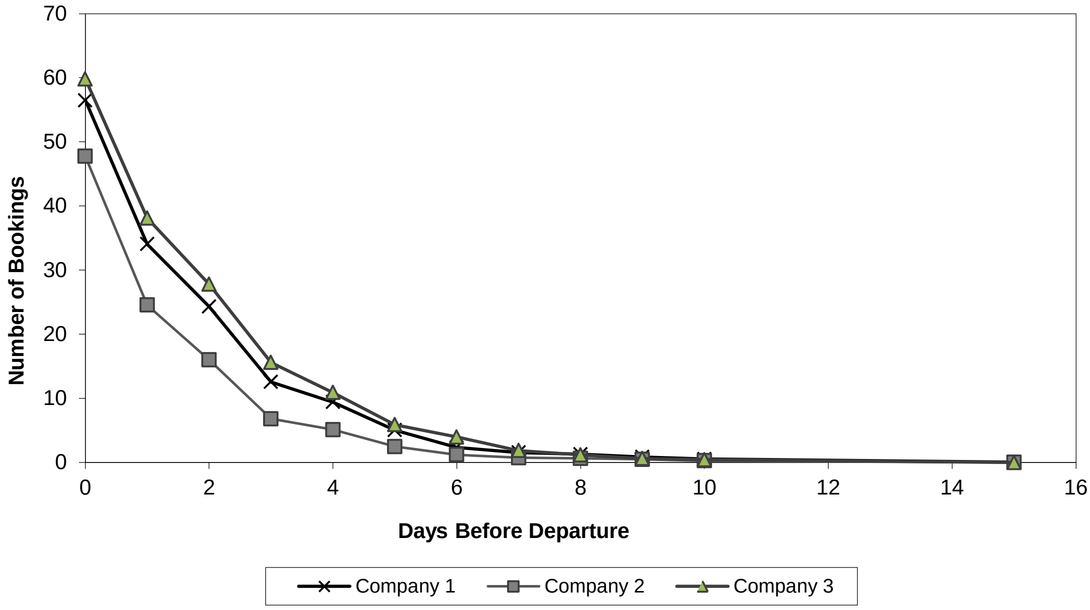


Figure 5 *Expected Cumulative Purchases for Each Company in Last Two Weeks of the Booking Period*

5 CONCLUSIONS

The analysis of the three competitor market in the previous section shows the clear advantage that a preferred player has over its competitors, highlighting the value of a good brand. With further analysis, drawing on the results of earlier work (Currie et al., 2008), we hope to be able to provide more insights into the optimal behavior of the middle player in these markets and which strategies enable it to increase its revenue and its market share.

We have not considered dynamic updating of prices here and so cannot evaluate dynamic strategies for dealing with competitors. As such, the competitors' prices are unaffected by the other players' strategies. In this particular scenario our main interest is how the customers behave and which airline gains the most revenue.

Our aim with this simulation model is to keep it as simple as possible so that the workings of the model are fully transparent. Therefore, although there are certain features that are needed to make the work more realistic, e.g. dynamic updating of prices, we are reluctant to include all the finer details of an RM system such as forecasting, manual intervention, public holidays, etc.

As the literature review shows, simulation has mainly been used for testing RM algorithms; however, it has the potential to be used much more extensively for exploratory analysis, such as that described here. This could act as a precursor to developing optimisation algorithms or be stand-alone work to gain insights into market behaviour.

The model as it stands could be used to evaluate different markets with differing numbers of competitors who behave in different ways. We are also very interested in examining different ways of writing the purchase probability and its dependence on the prices on offer. Here, we have assumed that the probability of purchase with a particular company is dependent on that company's price and the minimum price on offer in the market; however, it may be more realistic to consider the average price in the market, the maximum price or even the price of the known preferred competitor. We aim to present more of these results at the conference when we have had more time to test out the different scenarios.

REFERENCES

Anderson S, de Palma A and Thisse F (1992). *Discrete Choice Theory of Product Differentiation*. MIT Press, Cambridge MA.

Anjos M F, Cheng R C H and Currie C S M (2004). Maximizing revenue in the airline industry under one-way pricing. *Journal of the Operational Research Society* **55**: 535-541.

Anjos M F, Cheng R C H and Currie C S M (2005). Optimal pricing for perishable products. *European Journal of Operational Research* **166**: 246-254.

Belobaba P P (1989). Application of a probabilistic decision model to airline seat inventory control. *Operations Research* **37**: 182-197.

Belobaba P P and Wilson J L (1997). Impacts of yield management in competitive airline markets. *Journal of Air Transport Management* **3**: 3-9.

Ben-Akiva M E and Lerman S R (1985). *Discrete choice analysis: theory and application to travel demand.* MIT Press, Cambridge MA.

Bitran G and Caldentey R (2003). An overview of pricing models for revenue management. *Manufacturing and Service Operations Management* **5**: 203-229.

Currie C S M, Cheng R C H and Smith H K (2008). Dynamic pricing of airline tickets with competition. *Journal of the Operational Research Society* **59**: 1026-1037.

Elmaghraby W and Keskinocak P (2003). Dynamic pricing in the presence of inventory considerations: research overview, current practices, and future directions. *Management Science* **49**: 1287-1309.

Gallego G and van Ryzin G J (1997). A multi-product dynamic pricing problem and its application to network yield management. *Operations Research* **45**: 24-41.

Gorin T and Belobaba P (2008). Assessing predation in airline markets with low-fare competition. *Transportation Research Part A* **42**: 784-798.

Graf M and Kimms A (2011). An option-based revenue management procedure for strategic airline alliances. *European Journal of Operational Research* **215**: 459-469.

Kincaid W M and Darling D A (1963). An inventory pricing problem. *Journal of Mathematical Analysis and Applications* **7**: 183-208.

Kunnumkal S and Topaloglu H (2010). Simulation-based methods for booking control in network revenue management. In *Proceedings of the 2010 Winter Simulation Conference,* IEEE Piscataway, NJ: 1890-1897.

Lee S (1998). Modeling passenger disutilities in airline revenue management simulation. MSc thesis, MIT.

Levina T, Levin Y, McGill J and Nediak M (2011). Network cargo capacity management. *Operations Research* **59**: 1008-1023.

Littlewood K (1972). Forecasting and control of passenger bookings. In *Proceedings of the Twelfth Annual AGIFORS Symposium*, Nathanya, Israel.

Robbins H and Monro S (1951). On a stochastic approximation method. *Annals of Mathematical Statistics* **22**: 400-407.

Talluri K and van Ryzin G J (2004). Revenue management under a general discrete choice model of consumer behavior. *Management Science* **50**: 15-33.

Talluri K and van Ryzin G J (2005). *Theory and Practice of Revenue Management*. Springer-Verlag, Berlin.

van Ryzin G J and Vulcano G (2008a). Computing virtual nesting controls for network revenue management under customer choice behavior. *Manufacturing and Service Operations Management* **10**: 448-467.

van Ryzin G J and Vulcano G (2008b). Simulation-based optimization of virtual nesting controls for network revenue management. *Operations Research* **56**: 865-880.

Vulcano G, van Ryzin G J and Chaar W (2010). Choice-based revenue management: an empirical study of estimation and optimization. *Manufacturing and Service Operations Management* **12** 371-392.

Zhao W and Zheng Y (2000). Optimal dynamic pricing for perishable assets with nonhomogeneous demand. *Management Science* **46**: 375-388.

AUTHOR BIOGRAPHIES

CHRISTINE CURRIE is a lecturer of Operational Research in the School of Mathematics in the University of Southampton, where she also obtained her Ph.D. In addition she has an MPhys from Oxford University in Physics and an MSc in Operational Research from the University of Southampton. She is currently the Managing Editor for the Journal of Simulation and has just retired as co-chair of the Simulation Special Interest Group in the UK Operational Research Society. Her research interests include mathematical modelling of epidemics, Bayesian statistics, revenue management, variance reduction methods and optimization of simulation models. Her email address is christine.currie@soton.ac.uk and her web page is http://www.soton.ac.uk/~ccurrie/.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

DEVELOPMENTS IN THE QUALITY ASSURANCE OF GOVERNMENT MODELS USED TO SUPPORT BUSINESS CRITICAL DECISIONS

Mr Alan Robinson

Defence Science and Technology Laboratory
Portsdown Hill Road, Fareham
aprobinson@dstl.gov.uk

Mr Paul Glover

Defence Science and Technology Laboratory
Portsdown Hill Road, Fareham
peglover@dstl.gov.uk

ABSTRACT

Modelling is essential to the work of government. From providing the evidence to support major investment decisions to predicting the spread of pandemic flu, models underpin decisions which affect people's lives and have major financial implications. It is vital, therefore, that these models are fit-for-purpose. Experience with the Intercity West Coast franchise competition in 2012 underlined inter alia the importance of good quality assurance in the models that underpin government decision-making. One response to the issues arising from this competition was a cross-government review of the quality assurance of government analytical models (the Macpherson review).

This paper discusses the Macpherson review and the steps that are being taken cross-government to implement its findings. It concentrates on the key points that the review identified, in particular those with wide applicability across all modelling and analysis; and on the production of a "Rainbow Book" to capture best practice and guidance across Government.

Keywords: best-practice, guidance, modelling, analysis.

1 INTRODUCTION

The problems in the award process for the InterCity West Coast mainline franchise by the Department of Transport (DfT) in 2012 were well publicised at the time and led to the commissioning of two cross-government reviews. The first of these – the Laidlaw Review (2012) – addressed the specific issue of "what went wrong" in this particular case. In the expectation that at least part of the problem lay with the models and modelling used to support the DfT's decision-making a second parallel review was also set up. This multi-disciplinary review - led by Sir Nick Macpherson, Permanent Secretary at the Treasury – investigated the quality assurance of analytical models that inform policy across government. Although the Macpherson review was commissioned initially as a response to the InterCity West Coast mainline decision, its scope was deliberately pan-Government in order to consider any systemic issues and also to identify best practice that could be articulated and promulgated. Its final report was published in March 2013 – the Macpherson Final Report (2013).

This paper contains the following sections:

- Firstly, it briefly describes the conduct of the Macpherson review. It is worth emphasising that the review was multi-disciplinary, involving Operational Analysts/Researchers, Economics, Statistics, Policy Professionals, software experts and Social Science expertise as befits the very wide-ranging and diverse modeling stock that is used to underpin evidence-based decision-making across Government.

- Secondly, it considers the key issues addressed in the Macpherson review. This is presented in two parts. The first concentrates on some general principles and is followed by discussion of those key findings and recommendations from the review that are of pervasive interest to the modeling community.
- Finally, the paper describes the steps being taken to implement the review's findings. Note in this respect that, although both authors are from the Defence Science and Technology Laboratory (Dstl), which is an Agency of the UK Ministry of Defence (MoD), they have been involved in implementation at the cross-government level, within MoD and, more parochially, within Dstl itself.

2 THE MACPHERSON REVIEW – OVERVIEW

The review under Sir Nick Macpherson was supported by a senior level steering group, a cross-Government Working Group and a review implementation team. This latter team worked closely with all Departments across Government to identify *business critical models* (see further below) and associated approaches to quality assurance. In parallel, the team interviewed public and private sector organisations and professional bodies to identify best practice. Finally, drawing on the outputs from these information sources, it developed recommendations. In addition to the review final report of Mar 2013 (op cit) an interim report was also produced in Dec 2012 – the Macpherson Interim Report (2012).

Although the formal review is complete one of the key recommendations was that all Departments were asked to have plans in place to respond to the recommendations by summer 2013. It is also likely that a follow-up review, the form of which is yet to be confirmed, will take place to ensure that effective progress is being made on implementing the review's findings.

Many Departments, in particular those with a large diversity of modeling approaches such as MoD, set up their own working groups to support the review and subsequently to implement its findings. MoD declared around 60 models to the review around 13% of the total declared across Government; and also supplied key examples of its internal documentation, best practice and guidance to the central review team.

3 THE MACPHERSON REVIEW – KEY TERMS

The key terms used in this paper are discussed below.

Models. This paper uses the term model as a shorthand for the entirety of the modeling process, encompassing not just the model itself (usually instantiated in some form of software), but also the data used to drive it, the user(s) charged with using the model effectively, and any management regime or wider context within which the specific modeling is set. As a well-known aphorism states, "a fool with a tool is still a fool"; the model users are at least as critical as the software!

Model Life-Cycle. Typically, models progress through a four step cycle, albeit often with significant cycling between steps during the life of a model: scope, specify and design; build and populate; test; deliver and use. A final step might also be added – decommissioning – where the model reaches the end of its effective life and where any associated best practice and lessons learned can be distilled and passed on to subsequent modeling efforts.

Business Critical Models. The review team initially asked departments to submit details of all models that they considered to be business critical and the Quality Assurance (QA) processes applied to them. A model categorization was adopted based on the main sorts of decisions that the modeling supported, such as those used in forecasting, planning or in support of procurement and commercial decision making. The definition of which models were business critical was left to the judgment of departments; however, some outline criteria were suggested including the extent to which the model drives key financial and funding decisions, the extent to which it was essential to the achievement of the business plan, and the extent to which error could lead to serious financial, legal or reputational damage.

Quality Assurance. The review addressed QA regimes and options not as an end in themselves but as a means of providing the relevant assurance to the users of a model's outputs. Thus, while the presenting issues which led to the review were framed in terms of what Robinson S (2002: 107) has coined as *quality of the outcome* (referred to hence forth as *quality of outcome*), the majority of the focus of the review was with respect to the root cause of the particular problems encountered in supporting the Intercity West Coast main line franchise, namely *quality of the content* as framed by Robinson S (2002: 106-107) (referred to hence forth as *quality of content*). There was a much lesser focus on what Robinson S (2002: 107) has termed *quality of the process* (referred to hence forth as *quality of process*). The review was however careful to note that QA needs to be tailored to the nature of the decision which the modeling is supporting, such that too much QA can in some cases be as problematic as too little, particularly when considering the customer's needs and expectations, in terms of *quality of process* issues with respect to the imperatives of timeliness and cost. The QA processes considered ranged from those aspects concerning the *quality of content* issues that should be applied to all computer models, such as code checking, developer testing, use of version control and internal peer review, through to those aspects which as previously mentioned from some *quality of process* considerations, should be applied more selectively such as external peer review and full transparency (where the model and its associated data are published openly and subjected to scrutiny).

Fitness for Purpose. Evidence-based decision support within MoD is based upon a 'fitness for purpose' approach. A model that is wholly appropriate to support one particular decision may be totally inappropriate for another; as such the purpose at hand needs to be central. The fitness element can then assess whether the model matches the purpose, including concepts such as verification of the underlying model and the validation of its outputs against, for example, expert judgment, historical data or other models.

The key is to sufficiently assure *quality of content* issues while being mindful of *quality of process* issues and thus to match and tailor the review and assurance processes to the decision that is to be supported: In particular, making sure that the QA applied is proportionate given the risks and uncertainties that need to be managed. Further, it should be acknowledged that QA is a through-life process, seeking to deliver *quality of outcome* through sufficiently assuring *quality of content* and *quality of process* and not something just undertaken when final modeling results are delivered. Even though the ultimate 'fitness for purpose' decisions can only be taken at or near final delivery the pulse of decision making requires that relevant building blocks are put in place much earlier.

4 THE MACPHERSON REVIEW – KEY FINDINGS

Overall, the review found good signs in Departments' current practice on QA. These included the broad spread across departments of important basic *quality of content* techniques like code checking and verification, internal peer review, and the extent of internal guidance. Taken together, this indicated that key elements of QA are being widely and appropriately applied. Despite this, the review identified significant variation in the type and nature of QA used within, and between, departments. Much of this was to be expected given the differences in organisations' remits, and the levels of risk in question. However, it was not certain that this is always the case and the review's work highlighted the benefits of a more systematic approach to creating a work environment that expects high quality QA – including allocating clear responsibility for key models and how they are used, and giving specialist staff adequate time to manage QA effectively.

In broad terms, the review concluded that most components of best practice in QA fall under two headings: modelling environment, and modelling process. In particular:

- **Environment**: creating the conditions in which QA processes to deliver quality of the outcome can operate effectively, including through a culture that values QA and welcomes challenge, a well understood chain of responsibility and sufficient time for QA. It also requires adequate capacity, including specialist skills and sufficient time to conduct QA effectively.

- **Process**: establishing a clear process for every stage of the model life-cycle. This includes addressing *quality of process* issues through working alongside the customer to ensure there is a shared understanding about the purpose and any limitations of the model or models used. It is also about a systematic approach to make QA accessible, easy and comprehensive. This requires clear guidance on QA and clear documentation for every model.

Visible Senior Leadership. There should be visible leadership at the top of each organisation to create a culture that expects high quality QA and to support senior-level challenge. For example, this is discharged within Dstl by having a Board-level Modelling Champion supported by senior Subject Matter Experts within Dstl's individual Departments (of which Alan Robinson is one).

Model Senior Responsible Officer. A key recommendation from the review was that there should be a single specialist Senior Responsible Officer for each model ("Model SRO") through its lifecycle. Identifying such an individual – known, for example, in Dstl as a (software) model coordinator or SMC - provides a key point of knowledge about each specific model, including any strengths and weaknesses. This individual can provide advice as required about potential uses of their model(s) and can also be accountable for keeping key documentation in place and up-to-date.

Best Practice Sharing. As discussed further below, the review acknowledged that some Departments were stronger than others either with respect to (a) individual aspects of best practice or, more generally and thus more merit in sharing, (b) articulating and maintaining best practice and guidance across Government. Note, of course, that best practice does not necessarily imply a singular approach is appropriate in all circumstances; it is the effective and proportionate matching and tailoring of guidance to the particular case in hand that is crucial.

5 IMPLEMENTATION PROGRESS

Implementation is currently proceeding at three inter-linked levels, namely cross-Government, within MoD and within Dstl.

A cross-Government Working Group, led by the Government OR Service (GORS) and with strong inter-disciplinary representation, continues to operate to drive implementation and to develop and share best practice. In particular, a suite of best practice guidance with a draft title of the *Rainbow Book* is being developed. (This working title recognizes the multi-disciplinary nature of the guidance and is one of a family of "colourful" cross-Government guidance documents including, for example, a *green* book on appraisal and evaluation in central government and a *magenta* book providing guidance for evaluation). The *Rainbow Book* is envisaged as a modular suite of guidance with an initial delivery in Spring 2014, which can grow and be maintained as a central on-line resource. Its Chapters examine:

- Analysis, Modeling and Decision Making in Government
- Roles and Responsibilities in Analysis and Model Development
- Risks in Modeling and Analysis
- Quality Assurance, Verification and Validation
- Overview of Common Pitfalls

Most of the documents are being drafted by a distributed team using a Government-wide collaboration site known as Collaborate, which incorporates inter alia a wiki capability and an ability to collaboratively develop and share documents. The team are drawing heavily on the extant best practice and guidance that has been offered to the central team by Departments or is generally accepted in the broader literature, recasting that into a common format and then socializing the results to ensure broad sign up across the working level community, prior to formal sign off at senior level.

The core of the emergent *Rainbow Book* focuses on those issues which relate to *quality of content* and *quality of process* in equal measure so as to assure *quality of outcome* across the broad range of modelling applications across government. Annexes then provide guidance on the particular domains of model application highlighted by the Macpherson Final Report (2013), which in turn reference more detailed standards initiatives, relevant to specific fields.

One novel aspect of the development of the guidance has been a degree of crowd-sourcing over Collaborate to enable the writing team to reach out very broadly across the modeling community.

Dstl and MoD's prime contribution has been three-fold. Firstly, MoD and Dstl staff have been contributing to the guidance development to which each of the key individuals involved has bought c. 30 years of professional experience, much of it refined through teaching and leading seminars both in house and at partner Universities for c. 15 years. The role of the wider MoD and Dstl team has been to: participate in the shaping discussions, write key elements of the text and provide on-going review to the evolving documents. A parallel, collaborative venture referenced from the *Rainbow Book* – also hosted on Collaborate – called the Analytical Professional Network (APN) is also being used to socialise and develop MoD's inputs across the analytical professions as well as to help with the embedding of the best practice once finalised, enabling analyst professionals across government to work as part of a vibrant learning community. The APN provides analyst professionals with the capability to:

- Share the outcome of reviews into quality across the various specialisms, reflecting hard won experience concerning best practice;
- Raise emergent issues to crowd source insights from the wider body of analyst professionals, and;
- More broadly share emergent insight on best practice in their field.

Finally, a number of socialisation events have been held to alert staff to the importance of, and requirements for, appropriate QA; these aim to build on and strengthen an already strong QA culture.

The overall intention is that by contributing strongly to the central guidance the subsequent implementation within MoD and Dstl will be made much more straightforward as a minimum of tailoring to MoD/Dstl's specific needs will be required due to the buy-in that will already have been achieved.

6 SUMMARY AND WAY AHEAD

Models need to be fit for purpose to support the diverse and high impact decisions they underpin across Government. It is also important that their fitness for purpose can be suitably demonstrated and articulated so that those senior staff relying on the outputs of the modeling understand the strengths of the associated modeling along with any underlying weaknesses or residual uncertainty.

As recommended by the Macpherson review, an appropriate QA regime and associated best practice is being more formally articulated, building on extant best practice, and put in place across Government.

REFERENCES

The Laidlaw Review '*Report of the Laidlaw Inquiry: Inquiry into the lessons learned for the Department for Transport from the InterCity West Coast Competition*' HC 809 London: The Stationery Office https://www.gov.uk/government/publications/report-of-the-laidlaw-inquiry 2012

The Macpherson Interim Report ‘*Review of quality assurance of Government analytical models: interim report*’ PU 1403 HM Treasury https://www.gov.uk/government/publications/review-of-quality-assurance-of-government-models December 2012

The Macpherson Final Report ‘*Review of quality assurance of Government analytical models: interim report*’ PU 1404 HM Treasury https://www.gov.uk/government/publications/review-of-quality-assurance-of-government-models March 2013

Robinson S (2002) ‘General concepts of quality for discrete-event simulation’ *European Journal of Operational Research* 138 103-117

AUTHOR BIOGRAPHIES

ALAN ROBINSON has been working in Defence Operational Analysis for over 30 years. He is currently the Chief Scientist of the Policy and Capability Studies Department (PCS) of Dstl, which is an Agency of the Ministry of Defence, and has overall responsibility inter alia for the technical quality of the work delivered by PCS.

PAUL GLOVER received a BSc (Hons) in Archaeological Sciences from the University of Bradford in 1985. He then completed an MSc in Information Processing at the University of York in 1986 and has been working in Defence Operational Analysis ever since. Paul is currently a principal analyst in PCS Department at Dstl, with responsibility for the technical development and delivery of modelling and simulation methods and techniques.

Proceedings of the Operational Research Society Simulation Workshop 2014 (SW14)
B.S. Onggo, C. Heavey, T. Monks, B.Tjahjono and D-J. van der Zee, eds.

CONSIDERING VOLUNTEER BEHAVIOUR DURING RELIEF EFFORTS: A SIMULATION APPROACH

Abdelwahab Alwahishie

Department of Industrial Engineering
203 Freeman Hall, Clemson University
Clemson, SC 29634-0920
aalwahi@clemson.edu

Kevin Taaffe

Department of Industrial Engineering
130-A Freeman Hall, Clemson University
Clemson, SC 29634-0920
taaffe@clemson.edu

ABSTRACT

After a disaster, humanitarian organizations quickly respond by sending volunteers with different skill sets to the affected area in order to serve community's needs. Because these needs are changing over time, and because the number of volunteers arriving to and departing from the affected area is changing over time, inefficiencies in assigning volunteers to tasks arise. Even with well-behaved resources, it is a challenging assignment problem that contains stochastic demand and supply. However, volunteers are human, and their behaviour is influenced by their environment, in particular their work satisfaction. We consider a volunteer's flexibility to work on undesirable tasks and a volunteer's idle time waiting to be assigned as two elements that can reduce their remaining time on site. In this research, we propose an agent-based simulation model to study volunteer assignment policies based on scenarios that compare community need profiles, volunteer arrival/departure profiles, and volunteer attrition rates.

Keywords: agent based, volunteer, behaviour

1 INTRODUCTION AND MOTIVATION

Disasters can be classified as either naturally occurring (such as drought, earthquake, epidemic, flood, etc.) or manmade (such as hazardous material spills, terrorist activities, wars, etc.), and many of these disasters can often cause the loss of property, shelter, and basic provisions, as well as, in some cases, significant loss of life. In fact, according to Van Wassenhove (2005), an average of 500 large scale disasters kill about 75,000 people and affect a population of 200 million every year. Around the world, the number of natural disasters has doubled since the 1980's (http://menablog.worldbank.org/threat-natural-disasters-arab-region-how-weather-storm, accessed on 11/7/2013). For this reason, humanitarian organizations, or what we often refer to as Non-Governmental Organizations (NGOs), become more involved to help the affected populations. Many NGOs work in the affected areas in order to provide relief and begin the recovery effort. According to Jobe (2011), about 10,000 NGOs were working in Haiti at the time of the earthquake. One of the most valuable resources for NGOs is volunteers. About 109.4 million individuals participated in volunteer work, contributing 19.9 billion hours annually with a time value of approximately $225 billion (Shin and Kleiner (2003). Nevertheless, NGO's are still faced with daily questions about how to achieve the most from their volunteer workforce. While the volunteer participation numbers are staggering, there is also significant waste and inefficiency in the logistics of putting people to work quickly. volunteers are significant resource for the disaster, but they may be used inefficiently which may lead to health and safety issues. (http://www.seas.gwu.edu/~dorpjr/Publications/JournalPapers/jemarticle_fernandez.pdf, accessed on 11/15/2013). These organizations have questioned how they can retain more volunteers from event to event, while also increasing the impact the organization can have on the population-in-need during the disaster response. Volunteers give their time and effort without expecting payment, but if or when they feel dissatisfied, they have the ability to decide to leave the field anytime. Therefore, accounting for volunteer behaviour is a critical element in understanding how the individual volunteer may respond to his or her surroundings.

So far, little work has been done regarding volunteer assignment from the operations management perspective. In fact, we have not been able to find quantitative research that considers volunteer behaviour during the relief efforts. For this reason, the focus of this work is to introduce an agent based simulation model to capture volunteer behaviour and how it affects their time commitment, then we compare two policies for volunteer management that NGOs may use in practice.

The rest of this paper will be as follows. Section two provides a brief literature review about social work and operations management work. Section three introduces a detailed model description of the research problem. Section four presents the findings and results from the experimental testing and design. Finally, the conclusion and references are located in sections five and six.

2 LITERATURE REVIEW

Volunteer management has had great attention in the literature from a social work perspective when compared with the attention in the literature from an operations management work perspective. Social science research has focused on understanding what stimulates (positively or negatively) a volunteer to behave in a certain way. According to Connors (1995), when volunteers enjoy their work, they continue participating. However, poor interpersonal relationships will play a role in volunteers deciding to leave earlier than planned. In contrast, the operations management literature is much more sparse in its coverage of the topics regarding volunteer management, especially as it pertains to modelling how a volunteer responds to his/her environment. To address these areas, we expand on our research review of both the social work and operations manager trick.

2.1 Social work

The volunteer is the most valuable resource for humanitarian organizations. Therefore, understanding (1) what helps retain a volunteer and (2) what would cause the volunteer to quit earlier than original expected are key elements for decision makers to manage their volunteer workforce appropriately during disaster relief operations.

Volunteers may decide to leave early (or simply to leave) due to many reasons. First, volunteers quit because they have limited time availability. Family and career obligations often restrict the level of time a volunteer can donate to the relief effort. According to Hustinx (2010), about 55% of volunteers stop volunteering because there is a conflict with their job or studies, and 32% of volunteers stop volunteering because they want to have more time with their families. Second, volunteers stop working because *the assigned task does not match their interests/preferences*. When a volunteer is not assigned to the right position, he/she may feel dissatisfied and more likely to quit (Shin and Kleiner (2003)). Third, volunteers quit when the organization uses their time poorly. For example, according to Shin and Kleiner (2003), in a UPS study, 30% of volunteers quit for the reason of poor use of their time. Lastly, volunteers leave the field because they are not feeling appreciated or recognized and not feeling accepted by the volunteer group as a whole (Connors (1995), ELSTAD (2003), Hustinx (2010)). All of the above issues may affect the ability of the NGO to provide effective relief in an emergency situation, and this is one motivating point for our research. *By creating policy for how volunteer assignments are made, can we increase volunteer stays on-site while still accomplishing the work needed to be done?*

Quite logically, in order to retain volunteers, NGO’s should focus on avoiding the reasons that make them quit. In addition, volunteers continue working if they have already been volunteering in past relief efforts for a long time. As Elstad (2003) discussed, long-time volunteers have a stronger commitment in participating.

2.2 Operations management work

From an operations management point of view, the literature is rich with research on traditional labour practices. However, there is far less coverage on the topic of volunteer resource management. Traditional labour management has been reviewed by many authors in the literature. For example, Baker (1976) reviewed different mathematical models that have been developed for workforce scheduling with cyclical demand. In addition, Bechtold, Brusco and Showalter (1991) studied the performance of a number of heuristic methods of the labour tour scheduling problem. Moreover, Ernst, Jiang, Krishnamoorthy and Sier (2004) reviewed different applications, methods and models that have been done in rostering and personnel scheduling.

For volunteer scheduling, Gordon and Erkut (2004) developed a spread sheet-based decision support tool to generate shift times and schedule volunteers for the Edmonton folk music festival. They used integer programing formulation to handle the task preferences, with the goal of minimizing the number of surplus volunteers. In contrast, the cost of volunteer shortages was not clearly considered. In another study, Sampson (2006) demonstrated how volunteer labour assignment (VLA) problems are quite different from traditional labour assignment (TLA) problems. He considered the volunteer as a labourer with no cost; then he incorporated this difference into a goal programming model. In VLA, the goal was to minimize the total cost of assigning excessive volunteers, assigning too few volunteers, the actual volunteer assignments, and unsatisfied task demand. In addition, Falasca, Zobel and Fetter (2009) developed a multi-criteria optimization model to help in the assignment of volunteers to tasks. As in Sampson (2006), they reviewed the differences between a volunteer labour assignment and a traditional labour assignment. In another study, Falasca, Zobel and Ragsdale (2011) discussed the creation of a spread sheet multi-criteria volunteer scheduling model for helping a small development organization in a developing South American country. The goal of the model was to reduce the number of unfilled shifts, minimize the total scheduling costs, and minimize undesired assignments. This study is different from Sampson (2006) in that it considers that the volunteer labour cost is not negligible, such as travel expenses. Finally, Lassiter, Taaffe and Alwahishie (2013) developed a mixed integer programming model to help humanitarian organizations assign volunteers to different task skills. This model is the first one to consider volunteer training to help other volunteers with a different task skill level. They considered different probabilistic demand scenarios, and studied how the model responded to such uncertainties by assigning volunteers to different tasks. They found that preemptive training, even at the cost of not meeting a current need, can increase the ability to meet the anticipated work required in future periods.

3 METHODOLOGY

One approach to this problem would be to propose a mathematical programming formulation in which the periodic supply (i.e., volunteers and their skill levels) is allocated to the demands (i.e., the community's needs), where forecasts of volunteer arrivals and community needs are estimated and represented as unique scenarios. Initial research has been explored in Lassiter et al. (2013). While there is much to be learned on that research front, the main limitation is the manner in which an individual is characterized. The model cannot accurately assign and un-assign specific volunteers, while correctly tracking their progress in being trained to become a skilled volunteer. Even more importantly, we cannot monitor the individual volunteer's satisfaction with his or her assignments without adding indices to specify each volunteer explicitly. This would add an extreme amount of overhead to the formulation. While it could possibly be formulated in such a manner, this problem lends itself very readily to an agent based simulation approach. In this research, we developed the agent based simulation model in the AnyLogic simulation programming language.

In the agent based model, the main component is the agent itself, and because we are dealing with skilled and unskilled volunteers assignments, an agent class is created to represent the volunteer. Volunteer assignments and volunteer behaviour are controlled by state charts, which represent the possible states and transitions that a specific volunteer may experience. It is within the state chart that we can specify the policy that will govern how volunteers will be managed.

We consider two policies for modelling the behaviour of the volunteer and the assignment of volunteers to tasks. The entire state chart for the agent class of volunteers is presented in Figure 1 and Figure 3 to address the "No Training or Assignment Mismatching" policy or the "Training and Assignment Mismatching Allowed" policy, respectively. We will present each component of the state chart in detail in the following sections as they pertain to that policy.

3.1 Policy 1 – No Training or Assignment Mismatching

In this first model, the volunteer or agent is created upon arrival to the field. Key volunteer information to track includes: arrival time, skill level (skilled or unskilled), and time commitment. Upon arrival, the NGO assesses the skill level of the volunteer for placement into the appropriate group. The skill level is determined by a probabilistic condition and transitions that lead to the appropriate volunteer pool based on skill level (see Figure 1). We use a probability of 50% skilled and 50% unskilled. Each volunteer will have a planned time commitment that varies between four (4) and six (6) weeks. If the time commitment is reached, then the volunteer leaves the system. This can occur at the end of any week's worth of activity, whether actively working (AssignToHouseRepair or AssignToMealPrep) or waiting to be assigned (W1 or W2).

Once the above information has been specified, the organization assigns and sends volunteers to work only on tasks that match their qualifications or preferences. Skilled volunteers will be assigned to the task that require their skill (e.g., construction, roofing, home repair, etc.), while unskilled volunteers can be assigned to a task that does not require such labour skills. Examples of unskilled labour could include preparing beds and meals for people in shelters, or providing basic food and water needs to the community at large. The transition is used here to move volunteers to the field to start working on their tasks (AssignToHouseRepair or AssignToMealPrep). This step is done by checking the community needs, which are tracked by long term and short term needs variables that receive the input data from a table function (see Figure 2). In case there is no work to do, volunteers will begin waiting. For example, the W2 state captures skilled volunteers who are waiting for work. Being idle will affect volunteer behaviour, which can result in a volunteer's time commitment being reduced due to dissatisfaction. At the end of the week, volunteers will be assigned again according to the community's need. This is represented by paths that lead back to the skilled and unskilled volunteer pools for reassignment in the next week. In addition, there are new volunteers who will have just arrived and will be added to the pool of resources. Here we have a priority for the volunteers who were already working in the field. To do this, the transitions move volunteers back to be reassigned before the new volunteers are ready. Transition time plays an important role in determining which action will come first.

This process continues until the end of the study period, which we consider to be 20 weeks for our experiments. Performance measures are updated using events and calculated. For example, number of volunteers who left the field unsatisfied. That is, volunteer who decide to leave before they finish their commitment time.

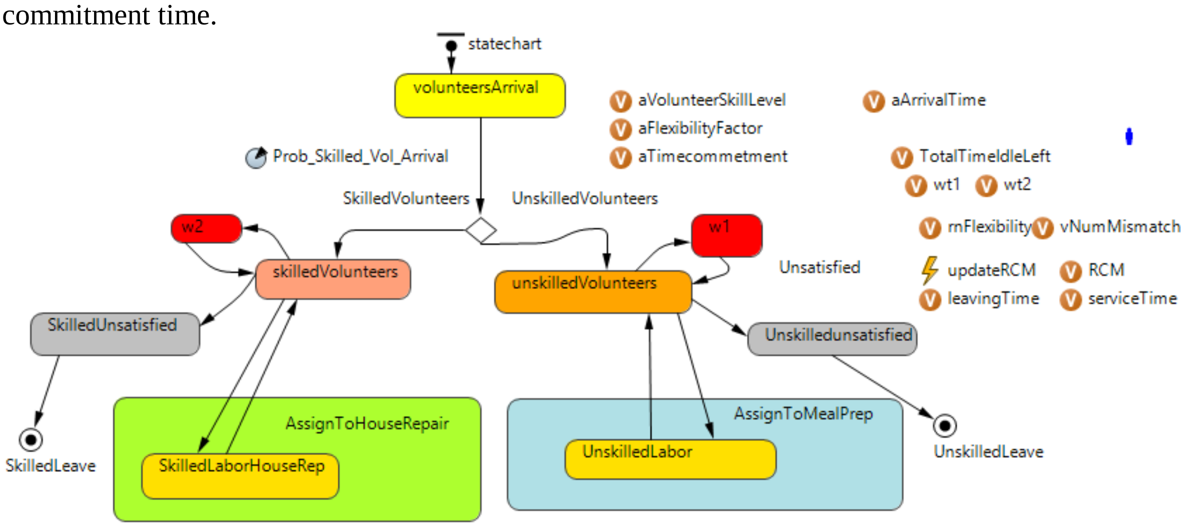


Figure 1 *Volunteer Agent State Chart: Base Policy model (no training or mismatching allowed)*

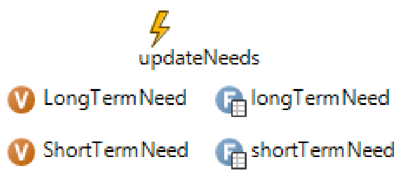


Figure 2 *Updating of short term and long term needs*

3.2 Policy 2 – Training / Assignment Mismatching Allowed

In this scenario, we consider the option of both (1) training unskilled volunteers to allow them to work on skilled tasks, and (2) allowing assignment mismatching (i.e., skilled volunteers assigned to unskilled tasks). As before, the volunteer arrives to the field, and we track arrival time, skill level, and time commitment. In addition, we now also include a flexibility factor (FF) with each volunteer. FF could possibly be determined during the initial interview. Volunteers who are willing to help no matter what jobs they perform are assigned a FF of 1.0. Those volunteers who can get dissatisfied with their assignments, and ultimately shorten their time commitment, would receive a value below 1.0. We consider two additional FF settings at 0.75 and 0.5 to denote those volunteers who are less flexible in their attitude towards their assignments. It is not clear how easy this information would be to obtain. However, it is important to measure the effect of having inflexible volunteers comprising the workforce in the field.

In this policy, we include both training and mismatching. That is, skilled volunteers who have no work can help the unskilled preparing meals when not enough unskilled volunteers are available in the shelters. On the other hand, unskilled volunteers who have no work can help complete skilled tasks, but they have to be trained while working. To account for this "training on the job," unskilled volunteers who are in training will be less efficient on skilled tasks than skilled volunteers or those volunteers who have completed training. After volunteers finish training, which has been set at two (2) weeks, they can work with full efficiency. Those volunteers will be categorized as trained volunteers, and they become a member of the "TrainedVolunteer" state (see Figure 3).

When assigning volunteers to tasks, not everyone gets his/her desirable task. The reason is that there is a variability in the demand and the volunteer arrival rates. As a result, some volunteers will net feel happy because the task is not what they expect, and will feel dissatisfied. In addition, for any type of volunteers, being idle with no work will affect their behaviour, and they may decide to quit.

The rest is the same as mentioned in Policy 1 except for the priority of who will be assigned first when the volunteer is available. The following are the actions that specify the rules governing Policy 2. Skilled volunteers are first assigned to tasks that require skill (i.e., skilled tasks). Unskilled volunteers are first assigned to tasks that do not require skill (i.e., unskilled tasks). If extra skilled volunteers are available, they can help unskilled volunteers if needed. If there is a shortage in the number of skilled volunteers, the following conditions must be followed (in the order in which they are listed):

1. If available, fully trained volunteers will be assigned to skilled tasks.
2. If available, partially trained volunteers will continue training and helping on skilled tasks.
3. If available, unskilled volunteers will start training and helping skilled volunteers on skilled tasks.

3.3 Reducing the Commitment Time – Idle Time or Mismatching

Volunteer dissatisfaction comes from both idle time and mismatching. We assume that one (1) week of being idle will reduce the volunteer's commitment time by one (1) week. For example, if a volunteer is willing to work for six (6) weeks, and in the first week he has no work, then he/she will leave after four (4) weeks.

Volunteers are also unique in their willingness to help on tasks that may not be aligned with their abilities or preferences. As stated in the prior section, we consider three different levels of flexibility: 0.5, 0.75 and 1.0. A higher flexibility factor (FF) indicates that the volunteer is less likely to become dissatisfied with his/her work even if the task is not a match to the original skill level of the volunteer. For volunteers with FF = 0.50, every time there is a mismatch their commitment time will be reduced by 0.5 weeks. For those with FF = 0.75, every time there is a mismatch their commitment time will be reduced by 0.25 weeks. Volunteers with FF = 1.0 will not be affected by the mismatch, so their commitment time will only be affected when they are idle.

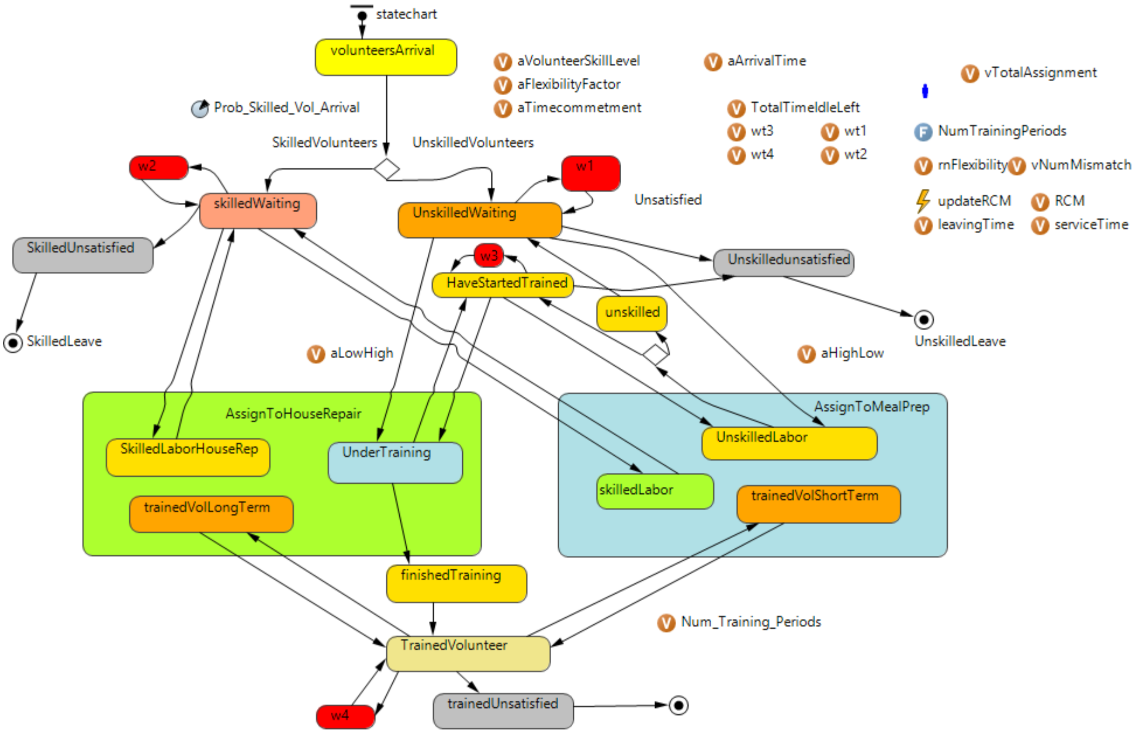


Figure 3 *Volunteer Agent State Chart: Training and mismatching allowed*

The following Reduced Commitment Time (RCT) equation depends on the flexibility factor for the arriving volunteer. For volunteers with FF = 0.5, the dissatisfaction equation is as follows:

$$RCT = max(0,aTimeCommitment-(wt1+wt2+wt3+wt4+vNumMismatch(1-FF))) \qquad (1)$$

where wt1 – wt4 denote the number of weeks that a volunteer sat idle in any of these states, vNumMismatch denotes the number of total mismatches incurred by this volunteer, and aTimeCommitment is the original commitment time for the volunteer. Thus, RCT can be compared against the elapsed time since the volunteer arrived, and a decision to leave or stay can be made.

4 EXPERIMENTATION

To test our model, four different demand scenarios (i.e., community needs) are considered. We consider a 20-week duration of the relief effort. In each week, needs are identified and categorized as short term (unskilled) or long term (skilled). The demand scenarios represent these short term and long term needs of the community. Needs are presented in terms of number of volunteers needed per week (see Figure 4).

The community needs for short term provisions are typically high immediately following the disaster event, and this is represented in all four scenarios. We then consider the short term needs to gradually dissipate. Notice that Demand Scenarios 3 and 4 have slightly lower, and consistent, initial short term needs than those shown in Demand Scenarios 1 and 2. For the long term need, we take into consideration the ability to begin addressing long term needs due to safety precautions. For Demand Scenarios 1 and 3, we assume that the NGO can immediately begin addressing the long term needs. In Demand Scenarios 2 and 4, we assume that it takes several weeks before the NGO can fully assess the extent of infrastructure repair and community redevelopment required. In these situations, it simply may not be safe to begin long term recovery until the area is determined to be safe. The impact is that we may see a higher and more sustained community need later in the recovery process.

In addition, we consider multiple arrival patterns for the volunteer workforce. Figure 5 presents the two volunteer arrival patterns under consideration. In Figure 5-a, volunteer arrivals are largest immediately following the disaster event, followed by a steady (linear) decrease of arrivals in subsequent weeks. In Figure 5-b, weekly volunteer arrivals remain constant throughout the relief effort provided by the NGO.

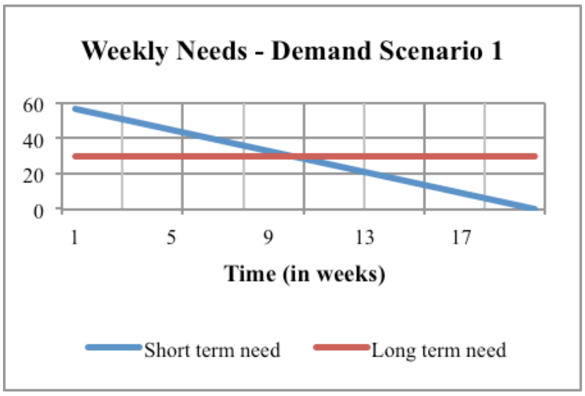


Figure 4-a

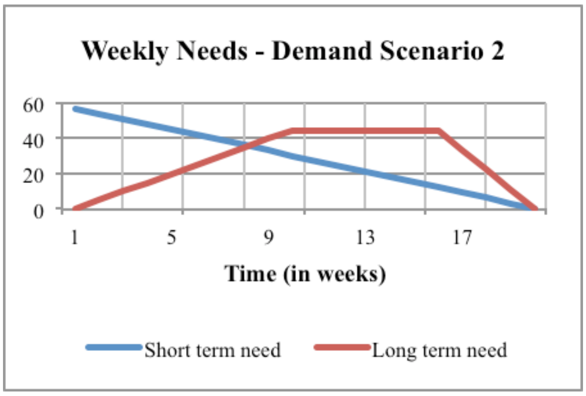


Figure 4-b

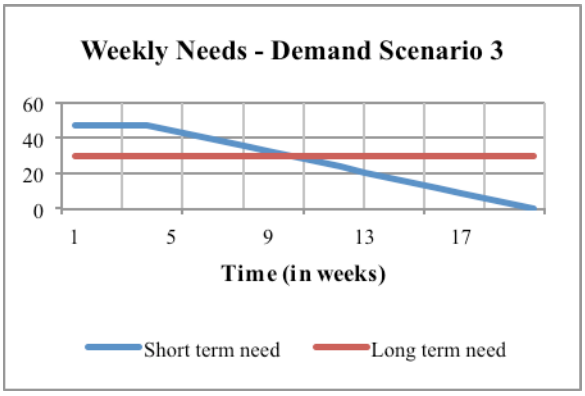


Figure 4-c

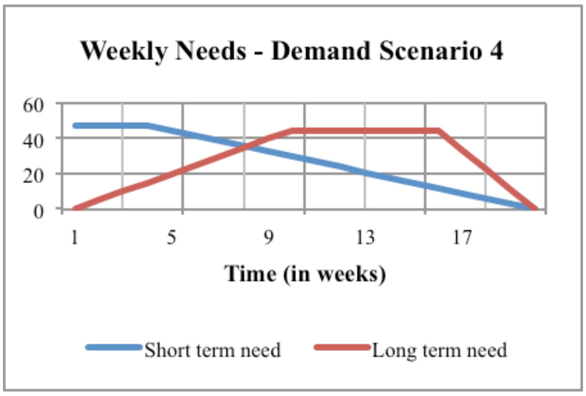


Figure 4-d

Figure 4 *Short-term and long-term community needs*

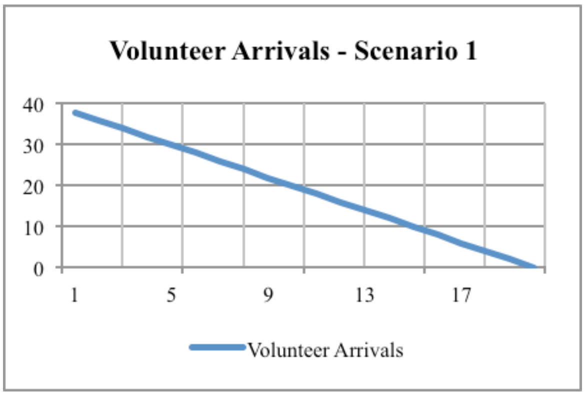


Figure 5-a

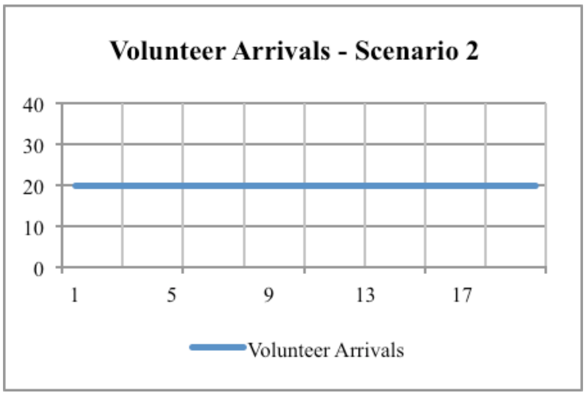


Figure 5-b

Figure 5 *Volunteer arrival patterns*

When volunteers arrive, they are classified as skilled and unskilled. A skilled volunteer can work on any task, but the unskilled volunteer needs training to be able to work on a task requiring some level of skill. While training, volunteers are actually helping the skilled labour but with less efficiency. We assume two (2) weeks of training for the unskilled labour in order to work with full efficiency.

All of the different combinations were run for a fixed number of 100 replications each. Table 1 summarizes the average and standard deviation of performance across all scenarios for each of the two policies – (1) not allowing training or mismatch, and (2) allowing training and mismatch. In addition, a comparison of means test was conducted for three of the performance measures. While volunteers in Policy 2 are now being mismatched, there is no statistically significant difference in percent of unsatisfied volunteers (p = 0.519) or in the total unmet need (p = 0.714). However, there is an 11.7% decrease in idle time (p =0.037).. Even though Table 1 indicates a slightly higher percent of dissatisfaction and fewer unmet needs for Policy 2, we cannot conclude any statistically significant difference due to the high standard deviations. These results provide motivation to further explore the benefit of providing training and allowing mismatching, but we also stress the importance of obtaining any data from the field. We are presently in contact with several NGOs but have not been able to secure estimates for the relief needs and volunteer profiles yet.

Table 1 *Policy 1/2 average (and standard deviation) values for the specified performance measures*

Policy	Total idle time (volunteer-weeks)	Percent of mismatched assignments	Percent of mismatched volunteers	Percent of unsatisfied volunteers	Total unmet need (volunteer-weeks)
Policy 1	**426 (45)**	**0%**	**0%**	**61.4% (5.4)**	**326 (160)**
Policy 2	**376 (40)**	**9%**	**22%**	**63.4% (6.3)**	**291 (201)**
t-test(0.05,7)	**Significant difference**	**-**	**-**	**Insignificant difference**	**Insignificant difference**

We also provide the scenario-specific results in Table 2 to explore the impact of the relief need (i.e., demand) and volunteer profile (i.e., supply). In Table 2, for any demand/volunteer/policy combination, the mean and the standard deviation under each performance measure were summarized, and comparison of means tests were conducted for all performance measures . The results show that there is a significant difference between Policies 1 and 2 in all scenario combinations except for the percent of unsatisfied volunteers under Demand Scenario 3 / Volunteer Scenario 1.

Table 2 *Policy 1/2 performance measures comparison by scenario*

Demand Scenario	Volunteer Scenario	Policy	Total idle time (volunteer-weeks)	Percent of mismatched assignments	Percent of mismatched volunteers	Percent of unsatisfied volunteers	Total unmet need (volunteer-weeks)
1	1	1	467 (20.2)	0	0	67.6 (2.7)	326.0 (62.5)
2	1	1	463 (20.7)	0	0	60.4 (3.2)	431.6 (124.3)
3	1	1	470 (18.4)	0	0	69.5 (2.6)	272.5 (61.0)
4	1	1	472 (21.5)	0	0	62.4 (3.0)	390.5 (121.8)
1	2	1	384 (16.9)	0	0	60.9 (3.2)	733.9 (93.5)
2	2	1	382 (17.6)	0	0	54.1 (2.9)	623.9 (129.5)
3	2	1	386 (17.9)	0	0	61.5 (2.9)	610.0 (85.9)
4	2	1	384 (19.2)	0	0	54.7 (2.9)	505.8 (128.0)
1	1	2	411 (20.3)	9.9 (2.0)	24.2 (4.8)	68.6 (2.6)	207 (38.9)
2	1	2	403 (19.8)	14.3 (2.2)	32.9 (7.2)	69.1 (2.4)	94.5 (47.4)
3	1	2	419 (19.3)	8.9 (1.8)	24.7 (5.4)	69.4 (2.5)	162.6 (30.7)
4	1	2	410 (22.2)	13.4 (2.0)	32.6 (6.0)	69.8 (2.1)	83.4 (46.0)
1	2	2	358 (17.3)	5.88 (2.4)	13.45 (3.8)	57.0 (2.6)	618.4 (59.4)
2	2	2	320 (20.1)	12.8 (2.3)	24.7 (6.1)	57.8 (2.5)	201.1 (32.1)

3	2	2	359 (16.3)	5.9 (2.3)	12.1 (3.6)	56.8 (2.6)	518.1 (54.4)
4	2	2	324 (19.0)	12.7 (2.3)	24.0 (5.2)	58.5 (3.2)	135.0 (22.2)

We also observe that, in most cases, the percent of the unsatisfied volunteers for Policy 1 is less than the percent of the unsatisfied volunteers for the same combination in Policy 2 , due to the fact that volunteers can be dissatisfied from mismatches in Policy 2. It is also interesting that we see the largest improvement in unmet needs (from Policy 1 to Policy 2) when either Demand Scenario 2 or 4 is exhibited. This indicates that by having a lower arrival rate of long term, skilled needs in the early weeks, there is additional time to train unskilled volunteers to be used for skilled needs before the demand for such skills becomes large. In addition, for the same combination, the idle time is smaller in Policy 2 since we allow volunteers to be mismatched and trained whenever needed (instead of remaining idle). We note that the current model is not predictive, i.e., we do not provide forecast information to the decision maker. This is another area of future research.

As another experiment, we chose to reduce the number of arriving volunteers per week by 25%. We were motivated to perform this test after our first tests resulted in such a large number of volunteer-weeks of idle time. However, while total idle time was about 50% lower for Policy 1 and 60% lower for Policy 2, the unmet needs more than tripled for Policy 1 and more than doubled for Policy 2. Operating with the current set of modeling assumptions, it will be difficult to eliminate both idle time and unmet needs. Again, there was no forecasting of decisions to be made based on future needs or arrivals. This leaves an interesting area of future research to explore.

We did see an increase in the percent of mismatched volunteers (and a subsequent reduction in idle time of 28% (p =0.013).) as compared to Policy 1, and Policy 2 does help meet the needs of the community better than Policy 1. In Policy 2, there was a 15.3% increase in volunteer dissatisfaction(p =0.045)., yet a 47% reduction in unmet needs resulted(p =0.024).. According to these results, a 1% increase in volunteer dissatisfaction lowers the unmet needs by 3.1%. Volunteer dissatisfaction is an important measure to consider, especially to retain them in the future, however, sacrificing some volunteer satisfaction may have a great impact in satisfying community needs.

Table 3 *Policy 1/2 average (and standard deviation) values considering a 25% reduction in volunteer arrivals*

Policy	Total idle time (volunteer-weeks)	Percent of mismatched assignments	Percent of mismatched volunteers	Percent of unsatisfied volunteers	Total unmet need (volunteer-weeks)
Policy 1	**210 (40)**	**0%**	**0%**	**39.5% (5.6)**	**1069 (322)**
Policy 2	**151 (30)**	**12%**	**37%**	**46.4% (5.7)**	**599 (333)**
t-test(0.05)	**Significant difference**	**-**	**-**	**Significant difference**	**Significant difference**

Table 4 is provided for completeness, as it contains all scenario-specific results from the experiment with a 25% reduction in volunteer arrivals. We do notice less volunteer dissatisfaction for both policies as a result of their being less idle time will be for both skilled and unskilled volunteers. Again, volunteers are more satisfied in Policy 1 than they are in Policy 2 since they only work on their preferred tasks. On the other hand, the unmet need is larger especially in the early weeks when volunteers are small in number. In Policy 1, a huge unmet need is observed due to the fact that volunteers, in this policy, work only on their preferred tasks. And, this may actually cause more community needs to arise than originally predicted or observed. The lack of providing enough services for community needs such as food, water, and basic medical supplies may cause a more severe situation for the community (and more work) in the future.

5 CONCLUSIONS

In this paper, we developed and presented a volunteer assignment and behaviour model for a better understanding of the impact of volunteer training and volunteer-to-task mismatching while providing relief to a community. We employed the use of an agent based model to provide a unique representation for the individual volunteer. This should be viewed as a first step in exploring the role of the volunteer during the relief effort, and how the NGO can best use and allocate its volunteer workforce.

Table 4 *Policy 1/2* p*erformance comparison by scenario, considering a 25% reduction in volunteer arrivals*

Demand Scenario	Volunteer Scenario	Policy	Total idle time (volunteer-weeks)	Percent of mismatched assignments	Percent of mismatched volunteers	Percent of unsatisfied volunteers	Total unmet need (volunteer-weeks)
1	1	1	233 (17.4)	0	0	45.5 (4.1)	665.2 (110.3)
2	1	1	251 (17.1)	0	0	42.0 (3.6)	1009.7 (206.2)
3	1	1	239 (15.1)	0	0	46.9 (3.8)	580.0 (106.1)
4	1	1	260 (19.4)	0	0	43.9 (3.6)	984.1 (217.4)
1	2	1	162 (15.7)	0	0	32.6 (4.1)	1353.4 (138.8)
2	2	1	177 (15.8)	0	0	35.3 (3.2)	1467.8 (289.46)
3	2	1	170 (16.1)	0	0	34.2 (3.9)	1177.5 (119.3)
4	2	1	184 (17.6)	0	0	35.8 (3.4)	1315 (249.1)
1	1	2	171 (18.1)	10.4 (2.3)	30.1 (7.0)	47.0 (3.8)	432.8 (56.0)
2	1	2	163 (17.8)	17.9 (2.1)	54.5 (9.2)	54.2 (3.8)	352.8 (108.5)
3	1	2	175 (17.0)	10.2 (2.1)	30.7 (6.7)	48.0 (3.8)	373.8 (57.0)
4	1	2	170 (17.6)	17.4 (2.1)	57.3 (10.4)	54.9 (3.2)	327.3 (98.7)
1	2	2	159 (12.6)	6.8 (2.2)	24.0 (7.6)	42.5 (3.9)	1198 (87.4)
2	2	2	102 (17.0)	17.1 (2.8)	57.9 (16.7)	41.9 (4.1)	609.1 (97.0)
3	2	2	163 (11.0)	6.8 (2.0)	23.3 (6.3)	42.9 (3.6)	1033.3 (86.7)
4	2	2	105 (15.3)	15.9 (2.3)	47.8 (17.0)	39.9 (4.1)	467.5 (99.2)

There were a few initial findings that were most interesting to include in a summary. The NGO would have more success in meeting the needs of the community if the need to work on long term, skilled tasks in the early weeks was minimal. This will allow additional time to train unskilled volunteers to be used for skilled needs before the demand for such skills becomes large. Volunteer dissatisfaction is an important measure to consider, especially to retain them in the future, however, sacrificing some volunteer satisfaction may have a great impact in satisfying community needs.

Finally, we note that the current model is not predictive, i.e., we do not provide forecast information to the decision maker. This is another area of future research.

REFERENCES

Baker K R (1976). Workforce allocation in cyclical scheduling problems: A survey. *Operational Research Quarterly,* , 155-167.

Bechtold S E, Brusco, M J & Showalter, M J (1991). A comparative evaluation of labor tour scheduling methods. *Decision Sciences*, *22*(4), 683-699.

Connors T D (1995). *The volunteer management handbook* John Wiley & Sons Inc.

Elstad B (2003). Continuance commitment and reasons to quit: A study of volunteers at a jazz festival. *Event Management, 8*(2), 99-108.

Ernst A T, Jiang H, Krishnamoorthy M., & Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research, 153*(1), 3-27.

Falasca M, Zobel C W, & Fetter G. M. (2009). An optimization model for humanitarian relief volunteer management. Paper presented at the *Proceedings of the 6th International ISCRAM Conference, Gothenburg, Sweden,*

Falasca M, Zobel C, & Ragsdale C. (2011). Helping a small development organization manage volunteers more efficiently. *Interfaces, 41*(3), 254-262.

Gordon L, & Erkut E (2004). Improving volunteer scheduling for the edmonton folk festival. *Interfaces, 34*(5), 367-376.

Hustinx L (2010). I quit, therefore I am? volunteer turnover and the politics of self-actualization. *Nonprofit and Voluntary Sector Quarterly, 39*(2), 236-255.

Jobe K (2011). Disaster relief in post-earthquake haiti: Unintended consequences of humanitarian volunteerism. *Travel Medicine and Infectious Disease, 9*(1), 1-5.

Lassiter K, Taaffe K., & Alwahishie A (2013). Improving volunteer productivity and retention during humanitarian relief efforts. *Industrial and Systems Engineering Research Conference ISERC,* San Juan, Puerto Rico.

Sampson S E (2006). Optimization of volunteer labor assignments. *Journal of Operations Management, 24*(4), 363-377.

Shin S, & Kleiner B H (2003). How to manage unpaid volunteers in organisations. *Management Research News, 26*(2/3/4), 63-71.

Van Wassenhove L N (2005). Humanitarian aid logistics: Supply chain management in high gear†. *Journal of the Operational Research Society, 57*(5), 475-489.

OR SW14

Worcestershire, UK

Proceedings

THE OR SOCIETY

The OR Society
1-2 April 2014

Worcestershire

ISBN 0 903440 56 3