Accurate Complex Systems Design: Integrating Serious Games with Petri Nets

Kirsten Sinclair[1], Daniel Livingstone[2]
[1] *Knowledge-Based Engineering Lab, Birmingham City University, United Kingdom*
kirsten.sinclair@bcu.ac.uk
[2] *Digital Design Studio, Glasgow School of Art*
d.livingstone@gsa.ac.uk

Abstract

Difficulty understanding the large number of interactions involved in complex systems makes their successful engineering a problem. Petri Nets are one graphical modelling technique used to describe and check proposed designs of complex systems thoroughly. While automatic analysis capabilities of Petri Nets are useful, their visual form is less so, particularly for communicating the design they represent. In engineering projects, this can lead to a gap in communications between people with different areas of expertise, negatively impacting achieving accurate designs.

In contrast, although capable of representing a variety of real and imaginary objects effectively, behaviour of serious games can only be analysed manually through interactive simulation. This paper examines combining the complementary strengths of Petri Nets and serious games. The novel contribution of this work is a serious game prototype of a complex system design that has been checked thoroughly. Underpinned by Petri Net analysis, the serious game can be used as a high-level interface to communicate and refine the design.

Improvement of a complex system design is demonstrated by applying the integration to a proof-of-concept case study.

Keywords: *Serious Games, Modelling, Systems Engineering, Analysis*

1. Introduction

In order to remain sustainable, organisations face the challenge of adopting regular changes in technology. Successful integration of technology into systems used to support operational processes is central to organisational strategy, yet difficult and unsuccessful [1].

Achieving desirable results requires gaining and providing a clear, accurate description of the proposed changes to people involved in an integration project. Presently, evaluation of proposed changes tends to be conducted late in the systems development lifecycle when the changes are made. This means shortfalls in understanding between people with different areas of expertise in the project (stakeholders) fail to be detected or are detected in the proposed design too late, i.e. when the changes have been physically made in real-life.

Visual forms of communication can promote shared understanding of a proposed design. Best practice in civil engineering, graphical modelling aims to explore and evaluate early design proposals before real-world implementation. Modelling is expected to provide a similar visual framework in systems engineering. Its use is anticipated to improve a design's accuracy and size while lowering the costs associated with building its real-world version too soon [2].

A graphical modelling approach used in systems engineering, Petri Nets, has been able to provide assurance that a design satisfies the original needs of stakeholders [3]. While use of Petri Nets improves description and checking of a system design [4], their form requires a level of familiarity that leads to a natural communication gap between project stakeholders. This gap in communication can negatively impact achievement of an accurate design for the proposed system.

Effectively communicating a design so that it is well understood among stakeholders is essential.

With their market share continuing to grow [5], serious games offer the potential for a player to achieve the learning outcomes the game was developed for and do so in an engaging way. Developers of serious games continually search for new ways to improve delivery of this learning experience. Mode of interaction can be viewed as essential to delivery of player engagement, particularly modes that offer a natural visual interface to improve player engagement and immersion [6]. In the new field of serious games, there are already multiple examples of serious games using realistic 3D graphics as an effective mode of interaction [7,8]. Less common are studies addressing accurate development of serious games, recognised as complex [9]. Winn et al [9] state the main concern is integrating non-linear game play, game content, and learning outcomes to deliver a serious game that meets its stakeholders' needs.

This paper reports on combining the complementary strengths of Petri Nets and serious games to achieve shared understanding and accuracy of a system design. The paper introduces the integration and initial results from a case study used to demonstrate its viability.

The paper is structured as follows. Section 2 discusses related literature. Section 3 describes the architecture of the integration. Section 4 reports on application of the integration to a case study. Section 5 presents conclusions and future research directions.

2. Related work

Coloured Petri Nets, like Classic Petri Nets, have a mathematical basis that enables content of the Petri Net model to be checked [10,11]. This checking can be done through interactive or automatic (non-interactive) simulation. It is not possible to thoroughly check modelled behaviour using only simulation. Thorough checking is achieved by calculating a reachability graph for the Petri Net model (also known as state space analysis). Graphical representations of Petri Nets can be created using toolsets such as CPN Tools [12,13]. Selection of CPN Tools' use in the case study of this paper is based on justification in [14].

As a tool to describe and check complex systems, Petri Nets have been applied [3,15] and evaluated [14]. Characteristics of a complex system and desirable features of a complex system design modelling tool are discussed in [14]. Considering [14], this work identifies an additional feature as necessary for a system design modelling tool, i.e. that of 'communicable to project stakeholders'. It could be argued that in spite of their usefulness as a tool to aid complex systems engineering, the reason the adoption of Petri Nets in industry remains low is due to their existing 2D graphical representation. This 2D representation only partially meets the need for a system design modelling tool to be communicable to project stakeholders. One potential approach to addressing this gap is to use graphically rich serious games as a prototyping mechanism for stakeholders to envisage a system design.

Serious game simulation has been used across multiple domains to envisage design. Examples include studies from [16-22]. Work centred on envisaging complex systems design includes [23,24]. All these studies have the objective of evaluating proposed changes in common. It was noted the work of [17,19] focuses on devices and not on how a system behaves as a whole. While the work of [16,18,23,24] is concerned with how a system behaves as a whole, there is no support for thorough checking of behaviour integrated into these approaches to ensure the serious game is accurate.

Research specific to developing serious game prototypes of Petri Net-modelled behaviour includes [25,26]. Of closest relevance, the work of [27,28] centres on prototyping user experience of ubiquitous computing system design by integrating a serious game and a Petri Net. Although all these studies are useful in terms of method and technology used to achieve a serious game prototype, they do not consider the integration of serious games and Petri Nets from a complex systems perspective. Complex systems design is concerned with combining component systems for a common purpose. Central to their successful engineering is understanding the interactions and information exchanges needed between these component systems and the associated timings, content, and sequencing. The integration of serious games and Petri Nets discussed in this paper takes this perspective into account.

The studies above also highlighted the use of both 3D application server and 3D gaming engines in the development and simulation of serious games. Based on comparison of the two platforms (Table 1), in this work, the Unity3D gaming engine [29] was selected as the tool to develop the serious game prototype of the system design modelled by the Petri Net.

Table 1. Required Serious Game prototyping capability.

Capability	Unity3D gaming engine	OpenSim 3D application server
Rich audio & visuals	Yes.	Less so.
Develop range of domain experiences	Yes.	Yes.
Content creation & publishing	Yes.	Focus on user-defined content.
Simulation of virtual world environment	Yes.	Yes.
Physics	Yes.	Via add-on physics engine.
Locally-served, low latency content	Yes.	Focus on dynamic, remote-served content.
Accessible toolset implementation	Yes, client-based & free (non-open source) version.	Less so, server-based. Open source & free to use.
Ease-of-use & support	Official & community forums.	Community forums.

3. Architecture of the integration

An overview of the architecture used to combine a Petri Net (developed in CPN Tools) with a serious game (developed in Unity3D) is shown in Figure 1.

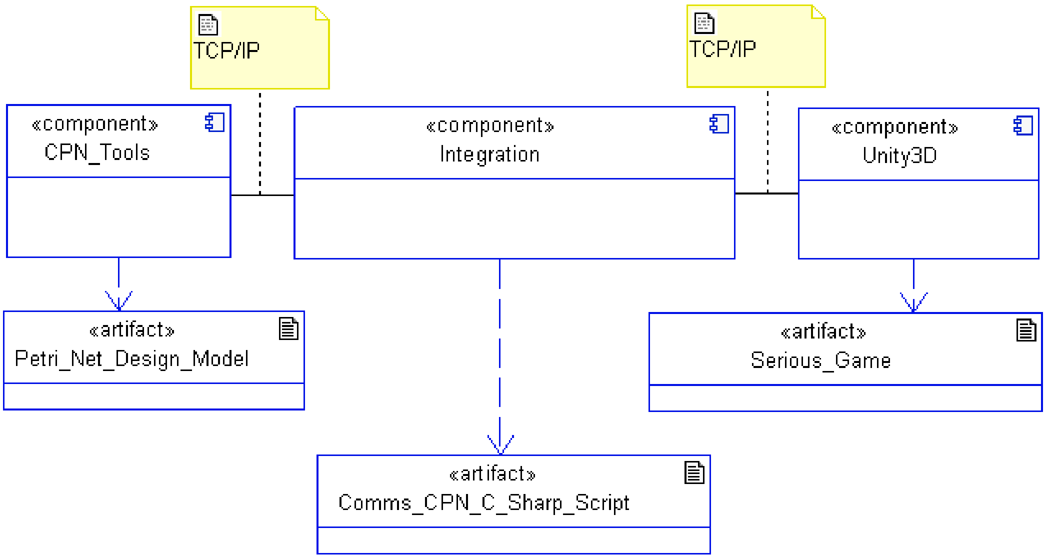


Figure 1. Logical architecture of the integration

The integration involves development of a Petri Net component and a serious game component. Both can exchange data using a TCP/IP networking component. The three components of the integration are described in more detail below.

3.1 Petri Net component

The integration uses the CPN Tools application to develop a Petri Net model of the proposed system design. This model of behaviour is built first and is process-based, i.e. it defines a sequence of actions with associated rules, information exchanges, and performers involved in the process. Analysis techniques of Petri Nets (simulation and state space analysis) are used to check the modelled behaviour thoroughly.

Once the Petri Net model has been checked, a serious game prototype of it is developed in Unity3D. Then the Petri Net is used to check that the developed serious game conforms to the behaviour it describes. Further actions, rules, and information exchanges are added to the Petri Net to manage this checking by:

- Setting-up a network listener service for establishing network communications with the game;
- Initialising the game environment in Unity3D;
- Receiving, processing, and/or sending data at required ‘behaviour checkpoints’ reached during simulation of the Petri Net.

Since this work aims to combine Petri Nets and serious games to improve design of complex systems, the main enabling characteristic of complex systems, i.e. the information exchanges needed between components [14], is focused on to identify behaviour checkpoints.

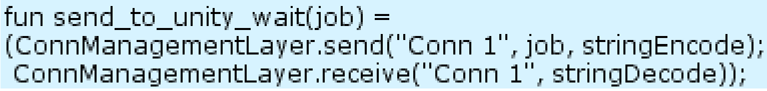
3.2 Serious Game component

The serious game is developed in Unity3D with the help of project stakeholders. 3D object creation and animation features available in Unity3D are used to build a virtual prototype of the system design modelled by the Petri Net. Location, viewing, navigation, and physics of each object specified by the game are managed by Unity3D when the game is played (simulated). The player (a project stakeholder) interacts with the game using Unity3D client software. CPN Tools compares the behaviour of objects in the game using the networking component. Unity3D uses the networking component to receive and process requests from CPN Tools to initiate or make updates to objects present in the game environment. It also uses the networking component to send data about game behaviour back to CPN Tools at required behaviour checkpoints identified in the Petri Net.

3.3 Networking component

A network connection between the Petri Net and the serious game is used to compare behaviour of the game prototype developed in Unity3D with that expected by the system design. It co-ordinates the gameplay with simulation of the Petri Net (when both are run in parallel).

Networking is implemented in CPN Tools through its pre-integrated Comms/CPN library [30]. This library allows Petri Net models developed in the tool to connect with external applications. Comms/CPN has accept, connect, receive, send, and disconnect functions [30] that can be called at points in the Petri Net model during simulation. An example combining use of the send and receive functions is shown in Figure 2.



```
fun send_to_unity_wait(job) =
(ConnManagementLayer.send("Conn 1", job, stringEncode);
 ConnManagementLayer.receive("Conn 1", stringDecode));
```

Figure 2. Example of the Comms/CPN library functions

In the game, networking is implemented by a script module in Unity3D that acts as a partner to Comms/CPN. This script is loaded automatically by Unity3D when the game loads. During simulation of the game, it provides functions for starting a connection, monitoring for data, receiving and processing data, and processing and sending data with CPN Tools.

When someone plays the game and triggers different behaviours, the script module in Unity3D notifies CPN Tools at required behaviour checkpoints. Short data updates about the behaviour are sent across the network connection by the game. In parallel, simulation of the Petri Net model can pause for and compare data received about the game’s behaviour to that expected by the design. Simulation of the Petri Net proceeds or halts dependent on the data received. If it proceeds, the Petri Net may send data to the game to initiate or make adaptions to objects in the Unity3D game environment for the player. The effect of data requests sent from CPN Tools is checked by the player of the game when they interact with the adapted objects.

4. Case study

To find out if and how the Petri Net and serious game integration can be used to address the complex systems design problem highlighted in Section 1, a small, proof-of-concept ‘heritage visitor experience’ from the Education domain was used. The objectives of the study were to demonstrate the feasibility of building the integration, develop a method for applying the integration, and identify the potential strengths and weaknesses of its application in describing and checking complex systems design. Although the study was small, it exemplified the following essential characteristics of complex systems (as highlighted by [14]):

- Multiple component systems;
- Multiple information exchanges between component systems.

The study indicated proposed new systems do not have to address an extended list of stakeholder needs before they involve interactions that are difficult for their stakeholders to fully envisage.

4.1 Overview of the required concept

Museum staff have identified a need for a new heritage experience where visitors can interact with the heritage subject and test their learning with a quiz. In the new system, a visitor arrives at the museum and is able to undertake learning by interacting with three heritage guides, two in no particular order, and then a third. The visitor views the information provided by all three guides and once they have done so, they are then ready to take the museum's quiz. This concept is summarised in Figure 3.

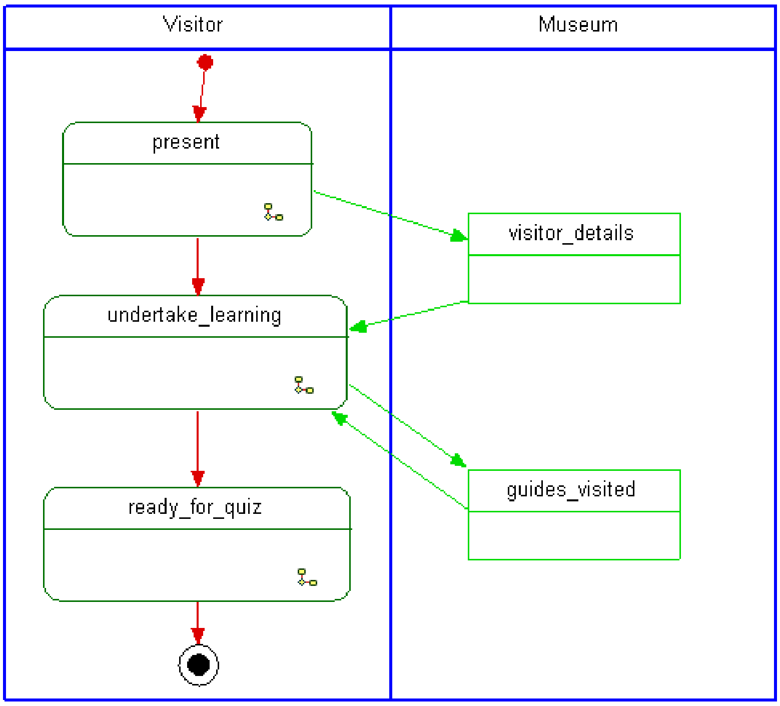


Figure 3. Overview of heritage experience concept

At this stage, the final solution to the initial concept presented by museum staff could be virtual (e.g. a serious game) or physical (e.g. staff interact with visitors at each wall-mounted guide), or a combination of both (e.g. audio-visual technology that visitors can interact with at each wall-mounted guide). This work is concerned with development of an accurate design proposal that can be used to describe what staff envisage so that the final solution selected and delivered for the museum is right in terms of functionality and cost.

4.2 Petri Net model of proposed system design

To accurately define what the new system should do, the case study started by building a Petri Net model from the overview presented in Section 4.1. The modelled process describes the expected behaviour of the system, outlining a particular timed sequence of actions, performers of actions, information exchanges, rules, and naming of actions and information. Required behaviour within the 'undertake learning' function was decomposed into the following main actions: 'visit a guide'; 'track guide visited'; and 'guide information read by visitor'.

For example, the ‘visit a guide’ action is part of the sequence of behaviour needed for a visitor to ‘undertake learning’. The rule for ‘visit a guide’ taking place is that the visitor has approached a particular guide they have not visited previously. Once ‘visit a guide’ has taken place, confirmation of the guide picked by the visitor is known. Information associated with this action can be suggested as: ‘visitor name’, ‘visitor details’, and ‘track guide seen’.

Actions derived from the overview were added as Petri Net transition elements. Information content derived from the overview was added in the form of Petri Net place elements. Colours (types) were defined in CPN Tools according to the information content needed at each place element. Compound or structured colour definitions were used for information that was to be updated by actions in the sequence. Figure 4 shows an overview of the proposed system behaviour described in the Petri Net.

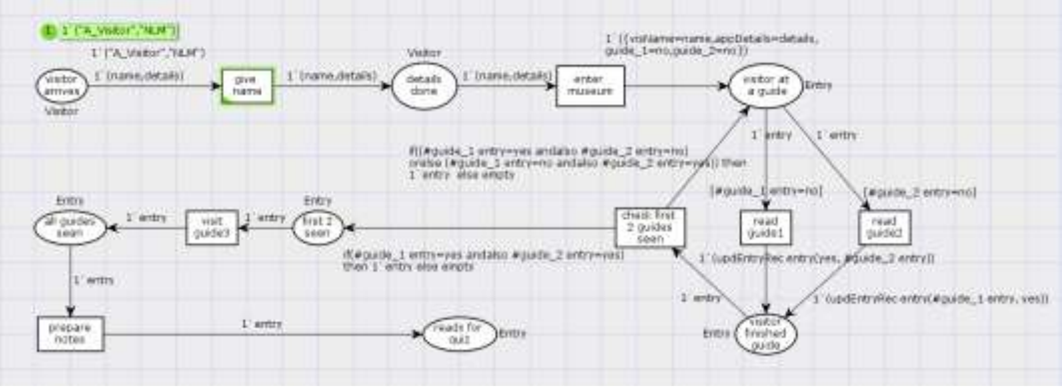


Figure 4. Petri Net representation of new system’s behaviour

4.2.1 Checking the Petri Net model

Once the initial Petri Net model of the system design was developed, it was checked for accuracy using Petri Net analysis techniques of simulation and state space analysis. Simulation of the Petri Net was used to provide confidence in the correctness of the sequence of actions within the model. It was also used to highlight points in the process where checking for accurate sequencing and content of information exchanges would be essential to achieving accurate design.

Simulation of an early version of the Petri Net (Figure 4) identified that when action ‘read guide x’ takes place, a token was produced for place ‘visitor finished guide’ and the process continues back to ‘visitor at a guide’, resulting in incorrect behaviour. The intended behaviour should be to test if the guide had been visited previously before action of ‘read guide x’ takes place, stopping a token from potentially continuing to loop through only one of the ‘read guide x’ transition elements. The token should pass through ‘read guide 1’ once and ‘read guide 2’ once. Conditions to check if the guides had been visited previously were added to ‘read guide 1’ and ‘read guide 2’ transition elements in a subsequent revision of the Petri Net.

State space analysis, used to complement simulation and provide a deeper level of checking, also highlighted the potential for an infinite loop. If this problem had not been detected using simulation, state space analysis would have alerted the Petri Net modeller to its presence.

4.3 Serious Game prototype

Following construction and thorough checking of the Petri Net model, its actions and information exchanges were then used to develop a serious game prototype of the modelled behaviour. Although the game was developed to be a realistic simulation of the new system described by the Petri Net, it was also a requirement for the game to be educational and entertaining. After playing the game, in terms of skills, stakeholders would be able to: discuss the new system from their perspective; and discuss aspects of the visitor who would eventually take part in the heritage experience. In terms of knowledge, stakeholders would be able to: list the rules of the new system; identify examples of the historical information to be included; identify examples of the environment where the heritage experience could take place; and identify the purpose of the heritage experience for the visitors likely to experience it.

Stakeholders would be motivated to appreciate the importance of envisaging designs for their new system and value the opportunity to discover more about needs for the new system from other stakeholders’ perspectives. Mechanics used in the game to realise these learning requirements were based on [31] and included a realistic graphical environment, selection/presentation/collection of information, navigation through/reaching certain points in gameplay, and use of on-screen prompts.

As indicated, the learning experience provided by the serious game was intended to be entertaining. To achieve this, entertainment aesthetics such as sensation, challenge, fantasy, and discovery were embedded in the game through use of mechanics [31] such as realistic graphics, audio, completion of tasks, a stakeholder taking on role of 'visitor', co-operation with other stakeholders, and use of on-screen prompts. These mechanics are introduced further below.

A virtual museum environment was selected for the prototype of the system design. Project stakeholders would evaluate the design of their envisaged heritage experience by playing the game prototype (taking the role of the visitor). They make their way through the virtual museum, locating and browsing the first two guides before the third so the accuracy of the heritage information provided by the guides and sequencing of actions could be checked. They would then undertake and check the quiz using the game prototype before gameplay ends.

For the game prototype, the role of the visitor was associated with the 'First Person Controller' object in Unity3D. Other objects needed to represent the heritage visitor experience in the serious game prototype were identified from the Petri Net and added in Unity3D and included: museum building and interior space (i.e. the area of gameplay), three guides (presented to the player/visitor as 2D pictures on the museum's walls [32]), and supporting objects such as lighting, textures, and audio.

With the game environment created, the sequence of actions and associated information exchanges in the Petri Net could then be represented in the game.

For example, 'visit a guide' in the Petri Net describes the actions, information exchanges, and performers involved in museum guide interaction that need to be prototyped by the game. Since communication and checking of the information exchanged in the design of the heritage system was essential, 'visit a guide' was prototyped by allowing the visitor (i.e. role played by a project stakeholder) to approach a guide and position themselves so they could:

- Engage a particular guide by mouse-clicking on it to choose it (at the start of the game, the guide is marked as interactive by setting a light object positioned above the guide to green, Figure 5);
- Confirm the historical information content presented by the guide (presented as text on-screen using graphical user interface functionality available within Unity3D scripting);
- Indicate the historical information has been processed (by mouse-clicking on a button to continue) before being able to move to the next part of the heritage visitor experience process in gameplay.

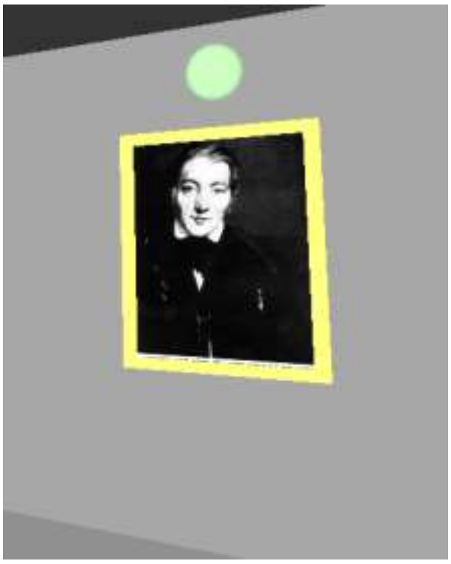


Figure 5. A guide object marked as interactive (light object above the guide set to green)

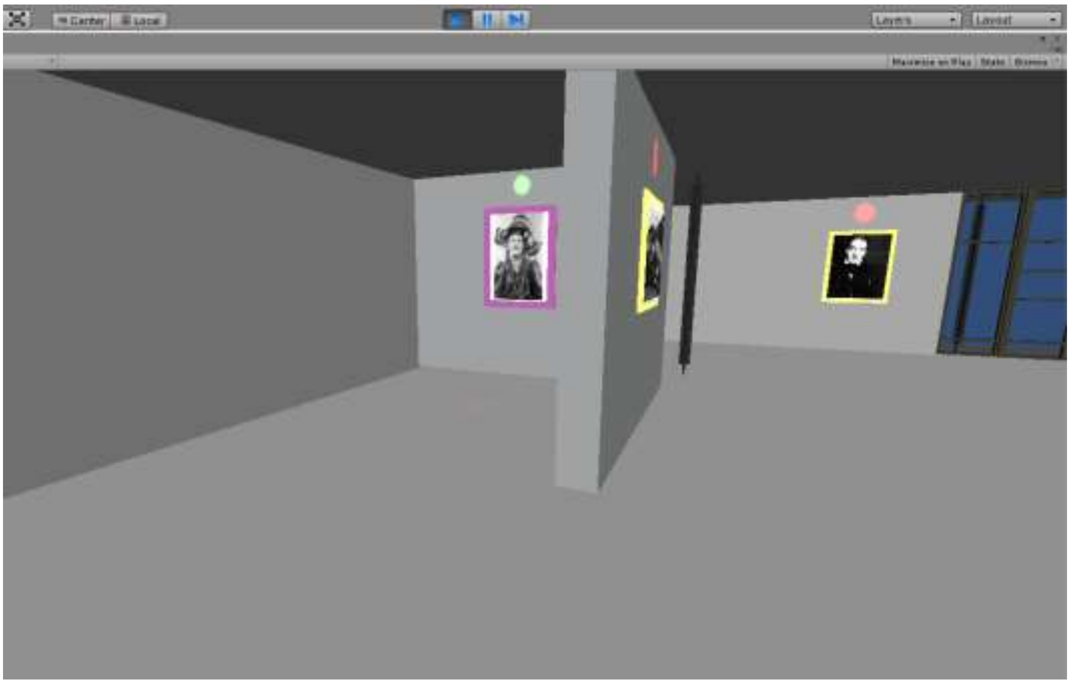


Figure 6. First two guides viewed by visitor (lights set to red), third guide now interactive

In the game prototype, the visit to the viewed guide was tracked (using Unity3D scripting) and once viewed, the guide was marked as no longer interactive by setting the light object above the guide to red. The Petri Net process describes that the third guide should not be viewed before the first two guides are seen by the visitor. This behaviour was realised in the game by positioning the third guide in an annex area of the museum interior. Access to the annex was blocked using a barrier object in the game. This barrier was removed using scripting following viewing of the first two guides by the visitor. The visitor was then able to approach and view the heritage information of the third guide (Figure 6).

Controlling access to objects through use of colour highlighting or hiding their presence during gameplay according to particular before and after conditions were the mechanisms used to represent and allow the player to check the sequence of actions described by the Petri Net (and in turn the design) through gameplay.

4.3.1 Managing the checking of the Serious Game prototype

Following development of the game prototype, further actions, rules, and information exchanges were added to the Petri Net. These ‘behaviour checkpoints’ (highlighted in red on Figure 7) were used to manage checking of the behaviour prototyped by the game. Through examination of the main behavioural actions of the Petri Net and their implementation in the game, behaviour checkpoints were added to manage networking with the game, initial game environment set-up, and interactions with guide objects.

The first behaviour checkpoint to be added to the Petri Net was action ‘setup listener’ (a sub-action of ‘setup Unity’ shown in Figure 7) for management of networking with the game. When the Petri Net is simulated and this action takes place, a network listener is set-up by CPN Tools. Simulation is paused in CPN Tools to wait for the game to establish a network connection. In the game, scripting was used to implement the corresponding action in the game, i.e. to establish a network connection with CPN Tools when gameplay starts. Once the game connects, simulation of CPN Tools resumes.

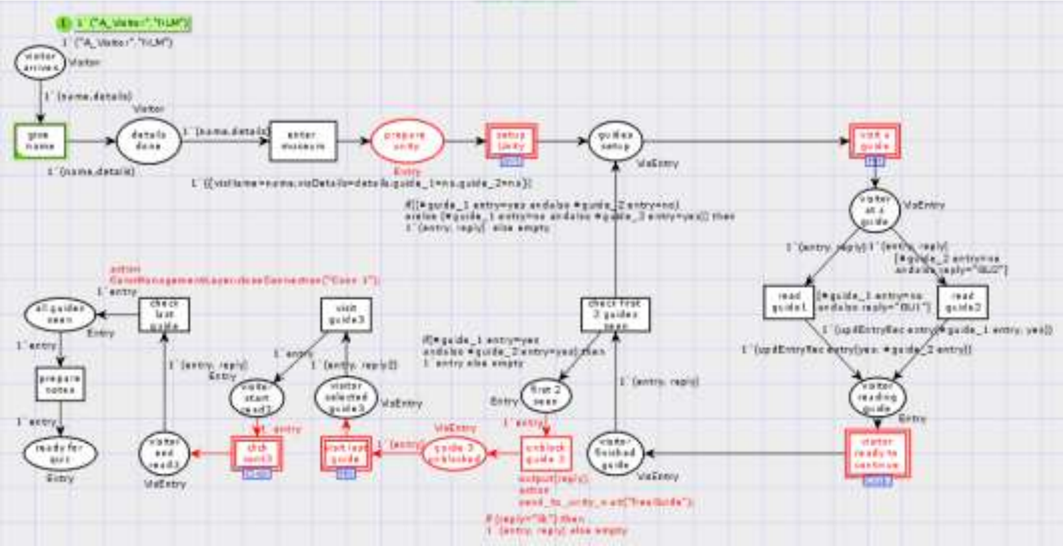


Figure 7. Petri Net with behaviour checkpoints added (highlighted in red)

An example of a behaviour checkpoint to manage ‘initial game environment set-up’ relates to the ‘visit a guide’ action. This involves correct set-up of guide objects in the game so that they are available and accessible for the player to visit and interact with. Before the ‘visit a guide’ action can take place in the Petri Net, action ‘configure guides’ was added to the Petri Net (a sub-action of ‘setup Unity’ shown in Figure 7). When this takes place, a ‘setup guide’ request is made to the game from CPN Tools (Figure 8). Unity3D’s networking script monitors for such requests, processes these, and adds or updates game objects accordingly. In this case, Unity3D adds the first two guides, sets them up as interactive (Figure 5), and notifies CPN Tools the guides are configured and ready in Unity3D.

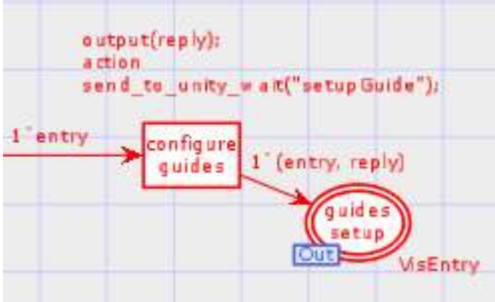


Figure 8. ‘configure guides’ action within the Petri Net

An example of a behaviour checkpoint to manage ‘interactions with guide objects’ relates to the Petri Net’s ‘visitor ready to continue’ action. When this action takes place in the Petri Net, simulation in CPN Tools is paused for the game to notify CPN Tools the player has completed processing the information content of a guide. This notification was implemented in the game using Unity3D scripting. When the player indicated they had processed the historical information (by mouse-clicking on a button to continue), an update was sent to CPN Tools.

It should be noted that requests made from the Petri Net to the game to add or amend objects did not delay gameplay. This will be monitored for future case studies where a greater quantity of objects may need to be added or updated in parallel. Gameplay effectively halts if the Petri Net encounters behaviour in the game that is not expected. This alerts stakeholders to the fact the Petri Net or Petri Net and game need correcting.

4.4 Checking system design using the Serious Game and the Petri Net

With the behaviour checkpoints in place for the Petri Net and its serious game prototype, gameplay and simulation of the Petri Net were started in parallel. This joint simulation was used by Petri Net and game modellers to check that:

- A network connection could be established between the game and CPN Tools;
- CPN Tools could request updates to objects and receive behaviour updates from the game;
- The game could receive requests from CPN Tools and send behaviour updates;
- Data passed between CPN Tools and the game was correctly synchronised.

The game could then be played by a project stakeholder while the Petri Net was being simulated. Doing so provided further checking of the heritage visitor experience captured within the game and within the Petri Net. The networking component of the integration handled the exchange of requests and behaviour updates between the game and the Petri Net at behaviour checkpoints.

This integration of the Petri Net and the game identified problems relating to incorrect ordering of information or missing information. These problems caused simulation of the Petri Net to halt and/or the stakeholder to query incorrect or missing behaviour in the game forcing the problem to be addressed. Examples of behavioural errors detected during simulation of the Petri Net and game included:

- Missing heritage information associated with a guide;
- Out-of-sequence interaction with a guide;
- Missing acknowledgment associated with selection of a guide for browsing.

Subsequent refinements to both the Petri Net and the game were made. To correct the out-of-sequence interaction with the third guide, the third guide object was hidden in the game. The requests to reveal and prepare the third guide object were moved to later in the Petri Net action sequence. Whenever a revision involved updating the Petri Net model, it was checked again using Petri Net simulation and state space analysis techniques.

In addition to identifying errors in the design, stakeholders were able to suggest changes to how the representation of objects could be improved for players during gameplay (and subsequently visitors using the new system). For example, checking of the heritage information provided by the guides (required to undertake the quiz) was achieved by enabling the player to interact with guide objects in the game. A player can mouse-click on a guide to reveal its associated heritage information. In the original version of the game, this interaction was implicit, i.e. it was a verbal instruction to the player that guides could be interacted with. During gameplay, the player suggested use of a colour marker positioned at the guide to re-inforce the possibility of interaction.

In this way, communication of the heritage visitor experience design described by a Petri Net was able to take place between project stakeholders. Having a stakeholder play a serious game version of the Petri Net provided a high-level interface for checking the design. Doing so led to detection of a number of errors. Table 2 presents an overview of the errors found in the case study using the combined components of the integration.

Table 2. Case study quantitative results.

Components simulated	Number of errors detected	Description of error
Petri Net and Serious Game	5	1. Missing heritage content associated with guides, e.g. additional historical text needed
		2. Heritage guide sequence of interaction incorrect, i.e. the guide picturing & describing the young girl needed to be the last guide visited
		3. Last guide could be interacted with out-of-sequence in gameplay
		4. Missing mouse-click acknowledgements relating to the status of the player's browsing of a guide, i.e. have they started or finished viewing it
		5. Errors in heritage content associated with guides, e.g. incorrect date information

From Table 2 it can be seen that playing the serious game while the Petri Net was simulated led to five behavioural errors being detected in the design of the heritage visitor experience.

Given the nature of these five errors, it may have taken longer or may not have been possible to find the same errors using Petri Net-based checking alone. As well as checking game behaviour, multiple revisions of the integration's components helped drive and evolve development of desirable behaviour for the new heritage visitor experience system.

5. Conclusions and future work

This paper has introduced the challenge of assuring accurate design of complex systems, integrating Petri Nets and serious games to address the problem, and applying the integration in a proof-of-concept case study. Conclusions made based on the results of this application are discussed below.

5.1 Conclusions

In summary, the main novel contribution of the work presented in this paper is:

- A checked serious game prototype for assuring accurate design of a complex system.
-

As well as identifying a method to apply the integration, the results from the proof-of-concept indicated its main strengths include:

- Petri Net analysis techniques for thoroughly checking the behaviour of the proposed system. The checked Petri Net model of the new heritage visitor experience informed subsequent development of its serious game prototype;
- Petri Net model to check the accuracy of the serious game prototype through networking of both models and identification of behaviour checkpoints;
- Playing the serious game enables project stakeholders with different areas of expertise to communicate and check the proposed design. Deeper involvement of stakeholders enables errors and gaps in the design to be highlighted during gameplay;
- It is an integration developed specifically from a complex systems engineering perspective. This work differs from current state-of-the-art [27,28] as it focuses on the interactions and information exchanges needed between components of a complex system and gaining shared understanding of these to improve accuracy of the proposed design.

The case study highlighted that problems found using the integration would not necessarily have been found using Petri Nets on their own. While the study confirms the integration makes a positive contribution towards accurate game and complex systems design, it also revealed the need to:

- Plan the required serious game environment with stakeholders so that a set of objects representative of their system under design are selected for the game prototype;
- Manage scalability of the Petri Net through careful planning of the hierarchy of the model to be built (to avoid state space explosion associated with large Petri Net models);
- Identify behaviour checkpoints between the Petri Net and the game.

The case study demonstrates the integration's potential capability to improve design of complex systems, justifying the pragmatic planning effort involved. To assess its feasibility further, work is being undertaken to apply the integration to a high-integrity complex systems design case study from a different domain, Defence.

5.2 Future work

Although fully functional, work to improve the integration continues. Areas to be investigated in the future include:

- Application of the integration to complex systems design case studies from other domains. This paper discussed its use in a short Education domain proof-of-concept case study. It is planned to confirm the integration's enhanced communication and analysis strengths by undertaking case studies with extended scope from other domains, e.g. Manufacturing;
- Multiple-player version of the serious game. The current case study's serious game was developed for one person to play (other stakeholders could view their gameplay). It may be useful for stakeholders to analyse the behaviour involved in a proposed system where multiple character roles can potentially interact to exchange objects and/or information and they are able to play the part of these roles;

- Integration of stereoscopy or motion capture technology. These technologies may be able to provide increased levels of realism in the serious game component, e.g. in safety-critical maintenance training where stakeholders need to check physical attributes such as torque or spatial distance in relation to the design of their new system.

Through serious games, project stakeholders involved in design of complex systems can take advantage of a realistic, interactive prototype of the design to gain shared understanding. Underpinned by the thorough checking capability offered by Petri Nets, the integration described in this paper can help develop both accurate serious games and accurate designs for complex systems.

References

[1] Karban, R., Weilkiens, T., Hauber, R., Zamparelli, M., Diekmann, R., & Hein, A, “Cookbook for MBSE with SysML”, MBSE Initiative – SE2 Challenge Team, 2011, available online at: http://mbse.gfse.de/documents/SE2PracticesAndGuidelines.pdf.

[2] Jorgensen, R., “Defining Operational Concepts using SysML: System Definition from the Human Perspective”, INCOSE International Symposium, Vol. 21, pp. 3005-3138, 2011. http://dx.doi.org/10.1002/j.2334-5837.2011.tb01307.x

[3] AbuSharekh, A., Kansal, S., Zaidi, A.K., & Levis, A.H., “Modelling Time in DoDAF Compliant Executable Architectures”, Proceedings of the 2007 Conference on Systems Engineering Research, pp. 1-10, 2007.

[4] Westergaard, M., “A Game-theoretic Approach to Behavioural Visualisation”, Electronic Notes in Theoretical Computer Science, Vol. 208, pp. 113-129, 2008. http://dx.doi.org/10.1016/j.entcs.2008.03.110

[5] IDATE Consulting and Research, “Serious Games: A 10 Billion Euro Market in 2015”, 2010, available online at: http://www.idate.org/2009/pages/download.php?id=147&idl=21&t=f_telech_actu&fic=518_Serious_Games_VA.pdf&repertoire=news/518_Serious_Games.

[6] Dow, S., Mehta, M., Harmon, E., MacIntyre, B., & Mateas, M., “Presence and Engagement in an Interactive Drama”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1475-1484, 2007. http://dx.doi.org/10.1145/1240624.1240847

[7] Hollander, A., Rose, H., Kollin, J., & Moss, W., “Attack of the S. Mutans!: A Stereoscopic-3D Multiplayer Direct-Manipulation Behaviour-Modification Serious Game for Improving Oral Health in Pre-teens”, Proceedings of the Stereoscopic Displays and Applications XXII, Vol. 7863, February 2011. http://dx.doi.org/10.1117/12.876759

[8] Lee, M.J.C., “Evasive Sidestepping as a Visual-Perceptual-Motor Skill: Implications for Anterior Cruciate Ligament Injury Prevention”, PhD Thesis, University of Western Australia, 2011, available online at: http://research-repository.uwa.edu.au/en/publications/evasive-sidestepping-as-a-visualperceptualmotor-skill-implications-for-anterior-cruciate-ligament-injury-prevention(fc3cd862-c32e-4090-874f-53b0e8a5b6f1).html.

[9] Winn, B., & Heeter, C., “Resolving Conflicts in Educational Game Design Through Playtesting”, Innovate 3, 2006, available online at: http://www.bsu.edu/eidm/fox/article3.pdf.

[10] Jensen, K., “Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use, Vol. 1, Second Edition, Springer-Verlag, 1997. http://dx.doi.org/10.1007/978-3-642-60794-3

[11] Jensen, K., “Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use, Vol. 2, Second Edition, Springer-Verlag, 1997. http://dx.doi.org/10.1007/978-3-642-60794-3

[12] Jensen, K., Kristensen, L.M., & Wells, L., “Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems”, International Journal on Software Tools for Technology Transfer, Springer-Verlag, pp. 213-254, 2007.

[13] CPN Tools, Home Page, 2015, available online at: http://cpntools.org/start.

[14] Sinclair, K., “The Impact of Petri Nets on System-of-Systems Engineering”, PhD Thesis, University of Durham, 2009.

[15] Lu, Z., & Levis, A.H., “A Coloured Petri Net Model of Distributed Tactical Decision Making”, Proceedings of 1991 IEEE International Conference on Decision Aiding for Complex Systems, Vol. 3, pp. 2035-2040, 1991.

[16] O’Neill, E., Lewis, D., & Conlan, O., “A Simulation-Based Approach to Highly Iterative Prototyping of Ubiquitous Computing Systems”, Proceedings of the 2nd International

Conference on Simulation Tools and Techniques, SIMUTools'09, pp. 1-10, 2009. http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5685

[17] Bylund, M., & Espinoza, F., "Using Quake III Arena to Simulate Sensors and Actuators when Evaluating and Testing Mobile Services", CHI2001, pp. 241-243, 2001. http://dx.doi.org/10.1145/634067.634210

[18] Brown, R.A., Eichhorn, D., & Herter, J., "Virtual World Process Perspective Visualisation", The Fourth International Conference on Information, Process, and Knowledge Management, eKNOW 2012, pp. 2-7, 2012.

[19] Nishikawa, H., Yamamoto, S., Tamai, M., Nishigaki, K., Tomoya, K., Shibata, N., Yasumoto, K., & Minoru, I., "UbiREAL: Realistic Smartspace Simulator for Systematic Testing", Proceedings of the 8th International Conference, UbiComp 2006, pp. 459-476, 2006. http://dx.doi.org/10.1007/11853565_27

[20] Rosendo, M., Buriol, T., deGeus, K., Scheer, S., & Felsky, C., "Towards the Development of a 3D Serious Game for Training in Power Network Maintenance", Third International Conference on Games and Virtual Worlds for Serious Applications, 2011, available online at: http://www.academia.edu/3016618/Towards_the_Development_of_a_3D_Serious_Game_for_Training_in_Power_Network_Maintenance.

[21] Wang, B., Haijiang, L., Rezgui, Y., Bradley, A., & Ong, H.N., "BIM Based Virtual Especifnvironment for Fire Emergency Evacuation", The Scientific World Journal, Vol. 2014, Article ID 589016, 2014.

[22] Torres, R.S., Biscaro, H.H., de Araujo, L.V., & Nunes, F.L.S., "ViMeTGame: A Serious Game for Virtual Medical Training of Breast Biopsy", SBC Journal on 3D Interactive Systems, Vol. 3, No.3, pp. 12-22, 2012.

[23] Chan, L.K.Y, & Lau, H.Y.K., "An Interactive Virtual Reality-Based e-Education System", Proceedings of the XIV World Congress on Information Technology (WCIT2004), pp. 1-10, 2004.

[24] Scacchi, W., Brown, C., & Nies, K., "Investigating the Use of Computer Games and Virtual Worlds for Decentralised Command and Control", Final Report, Grant #N00244-10-1-0064, University of California, Irvine, 2011.

[25] Carvalho, R.V., van den Heuvel, J., Kleijn, J., & Verbeek, F.J., "Coupling of Petri Net Models of the Mycobacterial Infection Process and Innate Immune Response", Computation 2015, Vol. 3, pp. 150-176, 2015.

[26] Guo, H., "Workflow Resource Pattern Simulation and Visualisation", PhD Thesis, Queensland University of Technology, 2014, available online at: http://eprints.qut.edu.au/65502/.

[27] Gomes, T., Abade, T., Harrison, M.D., Silva, J.L., & Campos, J.C., "Developing Serious Games with the APEX Framework", Proceedings of the Workshop on Ubiquitous Games and Gamification for Promoting Behaviour Change and Wellbeing, pp. 37-40, 2013.

[28] Silva, J.L., Ribeiro, O.R., Fernandes, J.M., Campos, J.C., & Harrison, M.D., "The APEX Framework: Prototyping of Ubiquitous Computing Environments Based on Petri Nets", In R. Bernhaupt et al (Eds.), HCSE 2010, Vol. 6409, pp. 6-21, 2010. http://dx.doi.org/10.1007/978-3-642-16488-0_2

[29] Unity3D, Home Page, 2015, available online at: https://unity3d.com/.

[30] Gallasch, G., & Kristensen, L.M., "Comms/CPN: A Communication Infrastructure for External Communication with Design/CPN", In K. Jensen (Ed.), 3rd Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, pp. 75-91, 2001.

[31] Lim, T., Carvalho, M.B., Bellotti, F., Arnab, S., de Freitas, S., Louchart, S., Suttie, N., Berta, R., & De Gloria, A., "The LM-GM framework for Serious Games Analysis", 2013, available online at: http://seriousgamessociety.org/download/LMGM_framework.pdf.

[32] SCRAN, Home Page, 2015, available online at: http://www.scran.ac.uk/.