2008

An Agent-Based Framework for Simulating Socio-Organisational Design of Large Projects

Ricardo Peculis
University of Wollongong, peculis@uow.edu.au

Derek Rogers
University of South Australia, derek.rogers@drderekrogers.com

Peter Campbell
University of South Australia, pcampbel@uow.edu.au

The success of large projects depends on how people associated in teams and organisations perform their tasks and cooperate to achieve a shared goal. Success depends on an effective socio-organisational design that defines organisational structures; human resources in the form of people's abilities, roles and responsibilities; and establishing processes and contracts that will drive measures of success. Predicting the behaviour of large projects is difficult and the performance of the socio-organisational design is likely to be validated only when it is put in to practice. Simulation offers a way to explore the likely behaviour of social systems without the drawback of negative effects on real situations. Agent-based models, in particular, can be applied to the simulation of organisations. This paper proposes an agent-based framework to simulate large projects, in particular those that involve the development of complex technical systems. The proposed framework is then applied to a simulation using Repast, an agent-based toolkit, to demonstrate how the framework can be applied to investigate feedback loops in the social system, and how these feedbacks affect the schedule, cost and quality of final products. The paper concludes discussing how agentbased models and simulation using this framework can help managers and decision-makers to acquire a better understanding of social systems, to explore socio-organisational designs of large projects with better chances of success and avoiding designs that would have higher risk of failure.

Keywords
framework, agent-based, design, projects, simulating, large, socio-organisational

Disciplines
Engineering | Physical Sciences and Mathematics

Publication Details
Peculis, R., Rogers, D. & Campbell, P. (2008). An Agent-Based Framework for Simulating Socio-Organisational Design of Large Projects. In E. Leigh (Ed.), SimTecT 2008 Conference Proceedings (pp. 291-296). Lindfield, Australia: Simulation Industry Association of Australia.

An Agent-Based Framework for Simulating Socio-Organisational Design of Large Projects

Ricardo Peculis[1]; Derek Rogers[2]; Peter Campbell[3]

[1]*Computer Sciences Corporation; SEEC University of South Australia, ricardo.peculis@postgrads.unisa.edu.au*

[2]*SEEC University of South Australia, derek.rogers@unisa.edu.au*

[3]*SEEC University of South Australia, peter.campbell@unisa.edu.au*

Abstract. The success of large projects depends on how people associated in teams and organisations perform their tasks and cooperate to achieve a shared goal. Success depends on an effective socio-organisational design that defines organisational structures; human resources in the form of people's abilities, roles and responsibilities; and establishing processes and contracts that will drive measures of success. Predicting the behaviour of large projects is difficult and the performance of the socio-organisational design is likely to be validated only when it is put in to practice. Simulation offers a way to explore the likely behaviour of social systems without the drawback of negative effects on real situations. Agent-based models, in particular, can be applied to the simulation of organisations. This paper proposes an agent-based framework to simulate large projects, in particular those that involve the development of complex technical systems. The proposed framework is then applied to a simulation using Repast, an agent-based toolkit, to demonstrate how the framework can be applied to investigate feedback loops in the social system, and how these feedbacks affect the schedule, cost and quality of final products. The paper concludes discussing how agent-based models and simulation using this framework can help managers and decision-makers to acquire a better understanding of social systems, to explore socio-organisational designs of large projects with better chances of success and avoiding designs that would have higher risk of failure.

1. INTRODUCTION

The success of large projects depends on how people associated in teams and organisations perform their tasks and cooperate to achieve a shared goal. Large projects are complex social systems where people interact with each other and with physical and conceptual objects that are created and modified in the course of the project. Success thus depends on an effective socio-organisational design.

Predicting the behaviour of such a design is difficult and the performance of the design is likely to only be validated when it is put in to practice. Interaction between intelligent actors produces adaptation and emergent behaviour that can create desirable and undesirable results. Managers and decision makers need to develop a better understanding of the behaviour of social systems and be supported by tools to assess the likely performance of large projects before the project starts. These tools should help to identify what socio-organisational design would have better chances of success at given conditions.

Computer simulation offers a way to explore the likely behaviour of social systems [4] and the impact of management in the life of organisations [11]. Agent-based models, in particular, can be applied to the simulation of social systems and have been shown to be appropriate to explore human behaviour [3]. Simulations produced from agent-based models can be applied to aid the socio-organisational design of complex projects [6].

This paper proposes an agent-based framework to simulate large projects, in particular those that involve the development of complex technical systems, such as software intensive projects and acquisitions. The framework expands BASP (Behaviour/Action Simulation Platform) framework [10] to adapt to Axelrod's framework for "harnessing complexity" [1] and the ACTS (Agent, Cognition, Task and Social environment) theory [2]. Like BASP, the framework comprises *agents* and *connections*. *Agents* are then expanded to *actors* and *artifacts*. *Actors* are active agents and represent people, while *artifacts* are physical and conceptual objects created and modified by the *actors*. *Connections* are the way to form teams and organisations, assigning tasks to *actors* and establishing dependencies between *artifacts*. A *connection* between *actors* establishes an *interaction*; a *connection* between *actors* and *artifacts* defines a *task*; a *connection* between *artifacts* establishes a *dependency*. An important aspect of *connections* is their activation, which determines when an *interaction*, *task* or *dependency* occurs. The activation of *connections* determines the sequence of events and is the mechanism that that can be used to simulate feedback loops in the social system.

The proposed framework is then applied to a simulation using Repast [9], an agent-based toolkit, to demonstrate that the framework is suitable to simulate large projects; to investigate feedback loops in the social system; and to understand how these feedbacks affect the schedule, cost and quality of final products. The paper concludes discussing how agent-based simulations using the proposed framework can help managers and decision-makers to acquire a better understanding of social systems, to explore socio-organisational designs of large projects with better chances of success and avoiding designs that would have higher risk of failure.

2. SOCIO-ORGANISATIONAL DESIGN

Socio-organisational design of complex projects involves defining organisational structures that delineate the interaction of people, teams and organisations; determining required resources in the form of people's abilities, roles and responsibilities; and establishing processes, contracts and statements of work that will drive measures of success.

The performance of social systems depends on how effective and efficient the system is in achieving its objectives. Effectiveness and efficiency are intrinsic characteristics of the system itself and how the system responds to external influences [8]. The performance of large projects depend not only on organisational structures, contracts and processes but is largely influenced by people's experience, knowledge and behaviour [6] [7]. Experience, knowledge and behaviour are difficult to assess and there is usually no reliable data available to effectively support project planning and management.

Large projects are complex social systems subject to adaptive and emergent behaviour that can move the project towards or away from its objectives. The socio-organisational design should address these aspects of social behaviour and create structures and conditions that would make the system flexible and robust to cope with unforeseen situations without compromising business objectives.

Strategic and business needs impose tight schedules, limited budgets and demanding quality factors, on top of pressure for high efficiency that would reduce costs, increase value for money and profit. The balance between efficiency and the flexibility that the system needs to handle what has not been accounted for is a fine line that managers have to deal with [8]. In practice, there is little room for error when designing the socio-organisational systems of large projects.

After the social and organisational systems have been established, project management is dedicated to manage the execution of the project against plans that define tasks, resources, sequence of events and expected results. Project management is highly dependent and constrained by the project's socio-organisational design. Success thus depends on an effective social-organisational design.

3. SIMULATING SOCIO-ORGANISATIONS

Computer simulation offers a way to explore the likely behaviour of social systems and agent-based models, in particular, can be applied to explore human behaviour in social systems [3][4].

In accordance with the ACTS[1] theory [2], organisations are collections of intelligent agents cognitively restricted, task oriented and socially situated. In the proposed framework, people are oriented to apply their knowledge, skills and perceptions of the situation in a social environment to a problem-solving group activity, comprising of expressing the need, formulating the problem, establishing constraints and finding and implementing a solution that satisfactorily meets the need.

BASP[2] [10] supports *agents* and *connections*, which have *variables* and *aspects*. The novel aspect of BASP is the separation between "behaviour" and "action". *Variables* and *aspects* are equivalent to Object Oriented *attributes* and *methods* and determine the "behaviour" and "action" of agents. According to BASP, "behaviour" establishes an "intention" of "action". Action is behaviour externalised. BASP defines *connection* as a connection between two agents and a *triple* as connections between three agents.

Socio-organisations are complex adaptive systems and Axelrod's framework for "harnessing complexity" [1] can therefore be applied for simulating socio-organisations. Axelrod's framework is based on three main components: variation, interaction and selection. In social systems variation, or variety, is the diversity presented in the system in the form of human resources that provide knowledge and experience to perform tasks that should move the system towards its goals. Interaction is what connects people with people and with the objects they manipulate. Selection is the process that determines strategies that are good and bad and consequently the ones that should be adopted and discarded. Axelrod's framework introduces the concept of "interaction activation" that establishes timing and determines the sequence of events.

4. PROPOSED FRAMEWORK

The proposed framework, shown on Figure 1, expands BASP to adapt to Axelrod's framework and the ACTS theory. Like BASP, the framework comprises of *agents* and *connections*. *Agents* are then expanded to *actors* and *artifacts*; *connections* are expanded into *interactions*, *tasks* and *dependencies*.

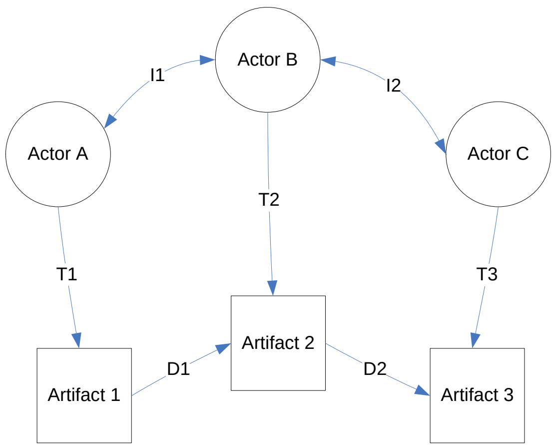


Figure 1: Proposed Agent-Based Framework.

[1] Agent, Cognition, Task and Social environment.

[2] Behaviour/Action Simulation Platform.

4.1 Actors and Artifacts

Actors are active agents and represent people. *Actors* have attributes that represent the *actor's* cognition through their knowledge, emotions and motivation. These attributes determine the *actor's* behaviour which results in actions. In the course of the simulation the *actor's* attributes may change, consequently changing its behaviour and actions.

Artifacts are passive agents and represent physical and conceptual objects that are manipulated, modified and created by *actors*. *Artifacts*, however, do not have cognition or behaviour. *Artifacts* are modelled with attributes that represent the *artifact's* ideal and actual values. The difference between the ideal and actual value corresponds to the distortion or error present in the actual *artifact*.

4.2 Interactions, Dependencies and Tasks

Actors interact with other *actors* and with *artifacts* through specific *connections*. *Connections* are the way to form teams and organisations, assigning *tasks* to *actors* and establishing *dependencies* between *artifacts*. A *connection* between *actors* establishes an *interaction*; a *connection* between *actors* and *artifacts* defines a *task*; a *connection* between *artifacts* establishes a *dependency*.

Interactions can happen formally, as part of the structure established by the socio-organisational design, or informally, through friendships and acquaintances. Either way, *interactions* influence the *actor's* behaviour and actions to a certain degree, depending on the *actors* and the *interaction* itself. The *interaction* defines the relationship between the *actors*, being of authority (superior/subordinate), peer or acquaintance.

Dependencies establish a dependency factor between *artifacts* and are the basis to define the *tasks* that the *actors* will perform. For products, as a collection of *artifacts*, to be effective and satisfy the need the dependency factors have to be maintained when the *actors* execute the *tasks* that are assigned to them.

Tasks define what the *actors* must do to create and modify *artifacts*. The framework proposes three types of *tasks*: *transformation*, *discovery* and *rework*. A *transformation task* creates *artifacts* given the input *artifact* and the transformation factor that represents the work the *actor* must perform. A *discovery task* is created and assigned to an *actor* for discovering distortions on *artifacts* already created. *Rework tasks* are in fact *transformation tasks* that modify *artifacts* to correct distortions discovered by *discovery tasks*. So that a *task* is performed without distortions the *actor* must possess the nominal level of knowledge required. *Tasks* also have a nominal level of effort that must be applied by the *actor*. If less than the nominal effort is applied by the *actor*, the *task* will be performed with distortions.

An important aspect of *connections* is their activation, which determines when an *interaction*, *task* or *dependency* occurs. The activation of *connections* determines the sequence of events and is the mechanism that creates feedback loops in the social system.

4.3 Activation of Connections and Feedback Loops

Connections have to be established and activated. A *connection* between an *actor* and an *artifact* is established when a *task* is created and it is activated when the *task* is assigned to an *actor*. *Connections* between *actors* are established when the *interaction* between the *actors* is created and it is activated when teams, organisations and acquaintances are established. *Dependencies*, the *connection* between *artifacts*, are always active; however, as *artifacts* do not have behaviour the effect of *dependencies* will occur when *actors* execute *tasks*.

The activation of *connections* can cause feedback loops in the system that impact on *artifacts* and *actors*. Figure 2 shows how feedback loops can occur. *Discovery tasks* create *rework tasks* that modify *artifacts* that through their *dependencies* impact on other *artifacts*, causing more distortions and rework.

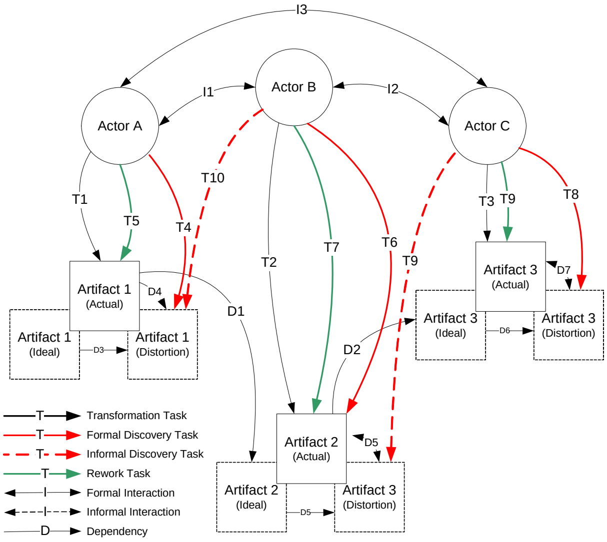


Figure 2: Feedback Loops.

Formal *discovery* and *rework tasks* included in review and test activities do not impact on the next product because the next product does not yet exist, and the rework occurs before the input to the next phase is baselined. Informal *discovery tasks* happen after the product is baselined when undetected errors are discovered during the next project phase and are fed back to the previous phase causing rework on input and output *artifacts* of the current project phase. It is not uncommon to detect defects on *artifacts* produced on earlier phases and reworking these *artifacts* will impact on *artifacts* that are connected to them through *dependencies*.

Feedback loops also occur on the social system and impact on *actors*. The activation of formal and informal *interactions* between *actors* influences *actors'* behaviour and actions. Cooperation is a positive form of interaction which can occur formally or informally. Cooperation increases shared knowledge and understanding, which in turn increases productivity and reduces errors and rework.

Another form of interaction that causes feedback loops in the social system is activated by the review process. When occurring on a constructive environment, reviews contribute positively for the project to achieve the desired goals. Reviews can also reveal problems that go beyond what is expected and may become aggressive. In these cases, reviews may trigger blaming and punishment, which increases management pressure, reduces group morale and productivity, which in turn move the project away from its objectives.

5. THE SIMULATION

The simulation was implemented with Repast 3.1 [9]. It aims to demonstrate the proposed agent-based framework and does not intend to be realistic or to reflect a real project. The simulation represents a software intensive project consisting of five phases:

- Phase 1: Operational Concept Definition,
- Phase 2: Functional Performance Specification,
- Phase 3: System Analysis and Design,
- Phase 4: Software Analysis and Design, and
- Phase 5: Software Development.

Each phase produces a product that will be input for the next phase. Distortions in one phase are propagated to the next [6]. The effectiveness of the project is determined by how well the final product highly dependent on software meets the needs that were the input for Phase 1.

5.1 Actors and Interactions

The simulation adopts the agent and social models presented in reference [6] to implement *actors* and *interactions*. *Actors* have their own knowledge, motivation, interaction style, role and responsibilities. The ideal *actor* possesses perfect knowledge and executes the *tasks* assigned to it without distortions. A more realistic *actor* has less than ideal knowledge and introduces distortions on *tasks* that it executes.

The *actor's* interaction style [3] determines its willingness to learn and cooperate with other *actors* [6]. Team performance and interaction with other *actors* influence the *actor's* motivation and its productivity.

5.2 Artifacts and Dependencies

The simulation adopts a vectorial task model that represents products and tasks within the acquisition space [5] [6][4]. Products are modelled as vectors in a Euclidian vector space, while tasks are transformation matrices that create other products. Each component of the product vectors becomes an *artifact*, and the transformation matrices provide the coefficients that define the dependencies between *artifacts*.

As an example, suppose that product P_1 is transformed into another product P_2 by the transformation matrix T (2x3), the *dependency* between P_2 and P_1 is represented by equation (1).

(1) $P_2 = P_1 . T$

If P_1 has two *artifacts* (p_{11}, p_{12}), P_2 has three *artifacts* (p_{21}, p_{22}, p_{23}) and t_{ij} are the coefficients of T, the *dependency* of P_2 *artifacts* on P_1 is determined by equations (2), (3) and (4).

(2) $p_{21} = p_{11} . t_{11} + p_{12} . t_{21}$

(3) $p_{22} = p_{11} . t_{12} + p_{12} . t_{22}$

(4) $p_{23} = p_{11} . t_{13} + p_{12} . t_{23}$

Each component of the transformation matrix is a *dependency* that connects a product's *artifacts*, and becomes a *task* that is assigned to an *actor.*

5.3 Tasks

Tasks are *connections* between *actors* and *artifacts* and require a nominal level of knowledge and effort to be executed. If the *actor* assigned to the *task* possesses at least the nominal knowledge and applies the nominal effort required, the *task* is executed without distortions. Otherwise, the *task* is executed with distortions that are proportional to the lack of knowledge and effort applied. The *task* may take longer if the *actor* decides to apply effort to learn and acquire the required knowledge.

Tasks can be *transformations*, *discovery* or *rework*. *Transformation tasks* transform input *artifact*s into other *artifacts*; *discovery tasks* aim to find distortions in *transformations* that have been completed; *rework* is another form of *transformation* that corrects distortions.

5.4 Development Process

The simulation implements a simple development process. A project phase starts when the previous phase has been completed and baselined. Each project phase receives an input *artifact*; executes all required

[3] Interaction styles are classified as constructive, passive or aggressive. Constructive behaviour fosters learning and cooperation; an aggressive *actor* will seek personal achievement through learning and without cooperation; passive *actors* will do what they are told and execute tasks with the knowledge they possess and without learning [6].

[4] Reference [6], "Using Simulation to Support the Design of Software Intensive Acquisitions", presented at *SimTecT 2007*, contains a summary of the Task Model [5] and presents the Agent and Social Models adopted in this simulation.

transformations to their completion; performs a quality control activity; corrects the distortions that have been discovered; baseline the output *artifacts;* and communicates the baseline to the next project phase.

After completing a development cycle comprising of *transformation*, *discovery*, *rework* and baselining, the project can perform more *discovery* and *rework tasks*, at the end of which a new baseline is created and communicated to the next project phase. The next project phase will accept the new baseline at the end of a development cycle.

6. SIMULATION RESULTS

The simulated scenarios represent the ideal and a more realistic case. The simulation starts with the definition of the ideal *artifacts*, *dependencies* and *tasks*. The results from the ideal scenario are used as the reference to assess the results obtained from the realistic scenario.

6.1 Ideal Scenario

The ideal scenario takes the representation of the ideal *artifacts* and their *dependencies* and creates ideal *actors* capable of performing the *tasks* without distortions and within the allocated time. The effort, duration and effectiveness of the ideal case are normalised and represented by 1.0.

The graphic on Figure 3 shows the results of the ideal scenario. The “Effectiveness” shows the contribution of each project phase to the final product as a whole. Each phase provides equal contribution to the final product. The work progress, or “Work Done”, progresses in accordance with the planned effort shown as “Work to be Done”. There are no errors and there is no need for *discovery and rework tasks.*

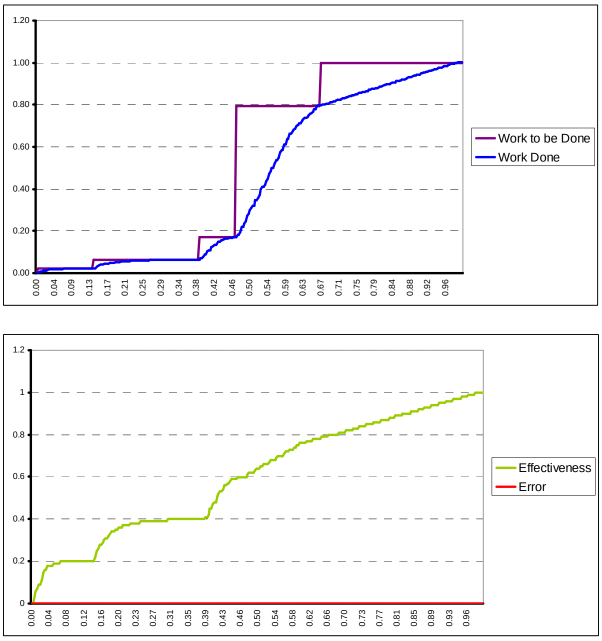


Figure 3: Ideal case.

6.2 Realistic Scenario without Learning

Figure 4 shows the results for a less than ideal case, where the same tasks are performed by teams and organisations that possess less than the required experience and knowledge. For this scenario, the simulation did not allow learning and every time a *task* is performed a constant percentage of the work produced comprises errors that have to be discovered and corrected. Although the quantitative result is only notional, the simulation reflects what happens on projects that struggle finding and fixing errors, which increases costs and extends schedule.

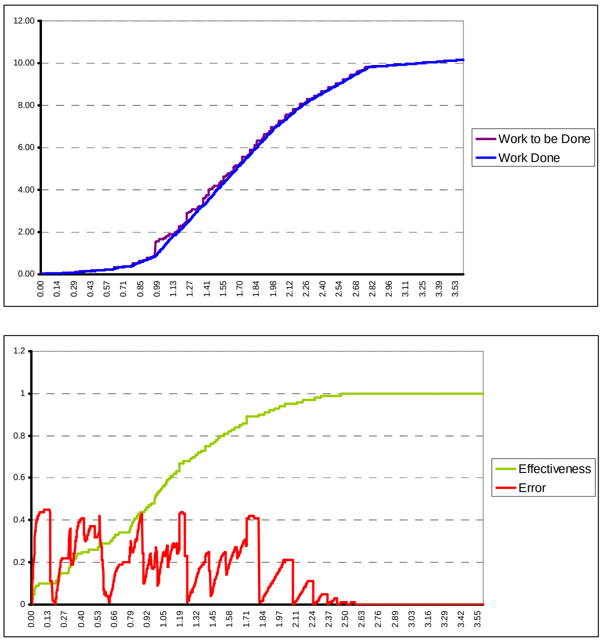
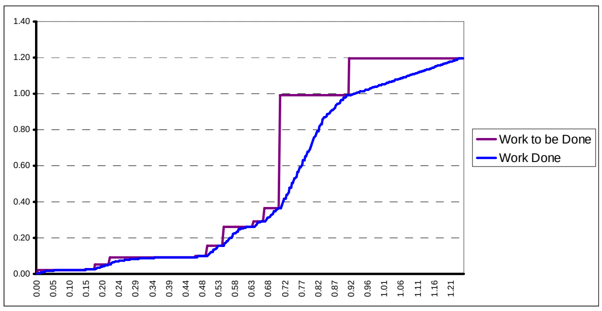


Figure 4: Less than ideal case without learning.

6.3 Realistic Scenario with Learning and Cooperation

Figure 5 shows the results for the same less than ideal scenario but with socio-organisational design that fosters learning and cooperation. In this case the *actors* learn as they perform *tasks* and cooperate transferring knowledge and experience to each other.



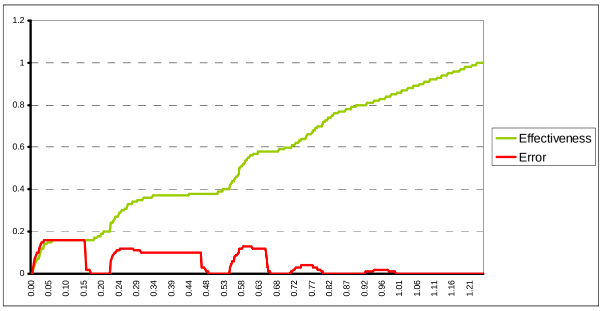


Figure 5: Less than ideal case allowing learning.

Through cooperation individual knowledge is shared and becomes collective knowledge, which is an emergent property of social systems. Every time a *task* is executed the *actors* learn and the errors produced are reduced. Learning and cooperation decrease the number of iterations, which reduces time and effort to complete the *tasks* and the project.

7. CONCLUSION

The success of large projects depends on the effective design of socio-organisations. Managers and decision makers need to develop a better understanding of the behaviour of social systems and be supported by tools to help in assessing the likely performance of the socio-organisational design before the project starts. Agent-based models and computer simulation can be used to explore the behaviour of social systems and the likelihood of success of large projects.

The proposed agent-based framework includes the basic elements that make the socio-organisational design of large projects. By modelling *actors* and *artifacts* and creating the *connections* between these agents the project can be simulated before the socio-organisational design is put in place and validated in practice. The activation of *connections* determines the sequence of events and can be used to simulate feedback loops that impact on *actors* and *artifacts*, on the social system as a whole, and influence cost, schedule and quality of the final product.

Although computer simulation provides ways to simulate social systems, modelling a real project is not simple. Obtaining reliable data to model knowledge and behaviour of the *actors*, how these attributes are influenced by *interactions* and how the *actor's* behaviour change over time is not easy. Large projects expand through multiple organisations and this kind of information, if in existence, is likely to be confidential. Organisations are also unlikely to provide information that may show they are not the best prepared for the job. If the data required to populate the model is not available, any expectation of obtaining quantitative results from agent-based simulations is compromised.

With few exceptions, agent-based models are not intended to produce quantitative results, and are more likely to be used to explore human and social behaviour and "*what if*" scenarios. At the beginning of large projects there is an expectation that everything will happen as planned: *the customer knows what they want; the provider is capable of delivering it; there are sufficient funds, resources and time to make it happen*. However, "*what if*" these statements are not correct? In that case, simulation offers ways to explore less than ideal scenarios and testing alternatives to compensate undesirable situations.

The benefits of simulation are not only the results the simulation provides. By investigating what is required to simulate the system a better understanding of the system itself is gained. The process of constructing a simulation of complex system can be as beneficial as, if not more, than the results that come from the simulation itself.

The proposed agent-based framework helps to simulate large projects and provides invaluable information of the building blocks that comprise their social-organisational design. By acquiring a better understanding of the social system of large projects, whether through simulation or by other means, managers and decision makers will be better prepared to develop designs with better chances of success.

REFERENCES

1. Axelrod, R., Cohen M. D. 2000, "Harnessing Complexity, Organizational Implications of a Scientific Frontier", Basic Books, New York, NY.

2. Carley, K. M., Prietulla, M. 1994, "ACTS Theory: Extending the model of bounded rationality", In Carley, K. M. and M. J. Prietulla (Eds.), *Computational Organization Theory*. Hillsdale, NJ: Lawrence Erlbaum Associates.

3. Davis, P. K. Henninger, A., 2007, "Analysis, Analysis Practices, and Implications for Modeling and Simulation", RAND National Defense Institute, http://www.rand.org/pubs/occasional_papers/2007/RAND_OP176.pdf.

4. Holland, J. 1995, "Hidden order: how adaptation builds complexity", Addison-Wesley Publishing Company, USA.

5. Peculis, R., Rogers, D., Campbell, P., 2007, "A Task Model of Software Intensive Acquisitions: An Integrated Tactical Avionics System Case Study", *12th Australian International Aerospace Conference*, Melbourne, Australia.

6. Peculis, R., Rogers, D., Campbell, P., 2007, "Using Simulation to Support the Design of Software Intensive Acquisitions", *Simulation Technical Conference, SimTecT 2007*, Brisbane, Australia.

7. Peculis, R., Rogers, D., Campbell, P., 2007, "Embracing Knowledge And Behaviour Management To Improve Performance Of Software Intensive Projects", *System Engineering Test and Evaluation Conference, SETE 2007*, Sydney, Australia.

8. Peculis, R., Rogers, D., Campbell, P., 2007, "Finding Better Strategies For Software Intensive Projects: The Efficiency Vs. Robustness Dilemma", *System*

Engineering Test and Evaluation Conference, SETE 2007, Sydney, Australia.

9. Repast, 2003, "Recursive Porus Agent Simulation Toolkit Overview", *http://repast.sourceforge.nel*.

10. Reynolds, W. N., Dixon, D., 2000, "A General Framework for Representing Behaviour in Agent Based Modeling", Complex Systems and Policy Analysis: New Tools for a New Millennium, Arlington VA, September 2000, RAND Corporation Science & Technology Policy Institute, *http://www.leastsquares.com/papers/rand2000.pdf*.

11. Stacey, R. D. 1999, "Complexity and creativity in organizations", Berrett-Koehler Publishers, San Francisco, CA.