

**TRANSPARENCY, CONSISTENCY AND MODULARITY OF STRATEGIC REASONING:
AN AGENT ARCHITECTURE FOR INTERACTIVE BUSINESS SIMULATIONS**

Rick van Krevelen
Martijn Warnier
Frances Brazier

Systems Engineering Group
Delft University of Technology
P.O. Box 5015, 2600 GA
Delft, THE NETHERLANDS

Alexander Verbraeck

Systems Engineering Group
Delft University of Technology
and
R. H. Smith School of Business
University of Maryland

Thomas Corsi

R. H. Smith School of Business
University of Maryland
3359 Van Munching Hall
College Park, MD 20742, USA

ABSTRACT

Interactive business simulations are widely used to explore and compare business strategies from both practice and theory. In many business simulations however, educators and researchers lack support in observing how the simulated actors operationalize their strategies, in validating whether operations have been aligned with the strategy, and also in (re)configuring available player and opponent strategies based on new theoretical or practical insights. This paper specifies requirements for a novel business simulation architecture that facilitates transparency, consistency and modularity of strategic decision making by simulated actors in interactive business simulations. A system architecture is proposed that integrates three components: an extensible agent middleware, a distributed simulation engine and a modular reasoning framework. How the architecture fulfills the three requirements of strategic reasoning transparency, consistency and modularity is illustrated in a use case of a business simulation game for supply chain management education.

1 INTRODUCTION

Games and simulations have long been applied as a tool for practice and training, from the ancient war games to today's complex (online) game environments. Since business games emerged early in the 20th century they followed the automation trend from paper to mainframes to home computers and web based simulation (Faria et al. 2009). Business simulations and games today educate management students in a variety of business strategies and tactics covering areas such as sales, finances and marketing. Typical advantages of simulation apply here, including relatively low cost training environments, no danger to participants, and avoiding ethical issues when things go wrong (Geurts et al. 2007, Pasin and Giroux 2011). Educators apply simulation games as a constructivist learning approach for their experiential or problem-based learning pedagogy (Lainema 2009, Thavikulwat and Pillutla 2010). Management research increasingly validates economic and computational models with interactive simulation to study human behavior in a complex system context. However, besides their advantages some of their challenges are now becoming clear.

One challenge of business simulations is their (lack of) transparency in linking business strategies and operations. Business games often claim to provide insight into business strategy and “prepare managers for the challenges they will face in the business arena” (van Houten 2007, p. 1), yet the simulated actors typically interact at the level of business operations rather than strategy. This means that participants in business simulations have to translate their envisioned strategies into whichever operations they see fit, possibly mediated by the usability of user interfaces or the overall time pressure.

Another challenge related to transparent strategy operationalization, is the lack of a disciplined relation between the chosen strategy and performed actions. Due to this opacity, strategic-operational consistency is difficult to verify in many business simulations, both interactive and not. For instance in the domain of Supply Chain Management, interactive simulations like the *Global Supply Chain Game* (GSCG) suite by Corsi et al. (2006) or the *RealGame* by Lainema and Hilmola (2005), although successful in engaging players with supply complexities, offer no clear means to verify that the selected strategy frames, guides, or constrains the player’s operational decisions. The player is free to make operational decisions that may or may not be consistent with the selected strategy and consistency may suffer. A similar challenge is found in non-interactive supply chain simulations for research, such as the *Trading Agent Competition for Supply Chain Management* (Sadeh et al. 2003) and many agent-based computational economics (ACE) experiments. Strategies are often described at a conceptual level with little support for communication, standardization or verification among researchers of their agent-based implementations.

This lack of discipline confounds the direct linkage between the selected strategy and its impact on player performance in the simulation, as well as on game facilitation. Since players’ operational decisions are not constrained by their selected strategies, educators, as game facilitators, are able to provide players with little support in understanding the impact of selected strategy on performance. Such guidance or scaffolding is critical for learning outcomes in the simulation-based exploratory educational setting (Bekebrede et al. 2005, Kirschner et al. 2006). The GSCG suite (Corsi et al. 2006) for instance does not support educators in ensuring that players actually follow the strategies they think or claim to follow. Without consistency between claimed strategies and actual performed operations, educators can not provide valid or appropriate feedback which make business games the effective learning tools they are claimed to be.

A third weakness in business simulations is the possibility to (re)design strategies composed of alternative, innovative tactics or policies. Some business simulations allow parametrization to provide some configurability of strategic reasoning, for instance van Houten and Verbraeck (2007) in the GSCG suite. However restructuring or completely replacing the automated actor strategies is typically not an option. Due to monolithic designs, business games often provide facilitators with a fixed reusable business rule-base and limited numbers of clients able to interact within the same simulation. Moreover, given the little regard for future maintenance and model redesign, business game development remains costly (Summers 2004, Tan et al. 2010). Modularity of strategic decision making may provide an answer, yet a robust and extensible architecture for large-scale business strategy simulation seems to be lacking. Considering the above issues, this paper investigates the question:

How to design a system that facilitates transparency, consistency and modularity of strategic reasoning by simulated actors in interactive business simulations?

The next sections are organized as follows. First, to arrive at requirements for a new system architecture for interactive business simulations, the observed shortcomings and related work is discussed. Following this section, a system architecture design is described that fulfills these requirements. The architecture’s supporting role in the domain of supply chain management and education is then illustrated in a use case. The paper concludes with a discussion on some opportunities and challenges that lie ahead for interactive business simulation architecture where strategies are operationalized transparently, consistently and extensibly.

2 SYSTEM REQUIREMENTS

Business simulations and games interactively simulate business environments with multiple strategic actors (corporations, banks, markets, consumers, etc.) in domains ranging from logistics operations to strategic management. In business games, players decide the actions of their designated actors so as to experience the effects of their envisioned strategies in competitive and dynamic business scenarios. While useful for engaging in business strategy, in this section some challenges are explored that lead to requirements for a novel system architecture for business simulation and games.

2.1 Transparent Strategy Operationalization

One primary concern in this paper is the apparent lack of transparency of how business strategies are operationalized within existing business simulations and games. Cornerstone aspects of business strategy such as sales, purchasing, transportation and logistics, have been researched extensively. However in capstone business simulations where the simulated overall business strategy incorporates a combination of these domain-specific policies, a transparent mapping between strategies and operations is commonly missing. Pasin and Giroux (2011) note that in many business games the facilitators must be well-informed to interpret the behavioral rules in order to provide appropriate feedback to individual player teams.

Comparing different mental models among strategic decision makers, Gary and Wood (2010) show that decision makers with more accurate mental models of the fit between different strategic options and the business environment, achieve higher performance outcomes (Gary and Wood 2010). It is therefore important that business simulation users (players and educators) observe valid causal relations that show how their simulated actors' operations actually led to the overall strategic performance. Yet there does not seem to be any capstone business simulation game that shows precisely how business (sub)strategies, concerning for instance the sales and stock policies, were translated into operations, such as sending orders and receiving shipments, and in turn led to the overall business performance.

Furthermore, empirical results in business game research suggest that performance and learning do not covary (Gosen and Washbush 2004). Pasin and Giroux (2011) thus started to measure learning not by evaluating simulation performance but rather analyzed each of the students' decisions to identify specific strategically 'wrong' decisions or 'mistakes' that their logistics management simulation game was intended to reduce. Measuring learning by tracking the number of strategic mistakes is new in management simulation games (Pasin and Giroux 2011) and requires transparency of these inconsistencies. There is therefore a clear need for support of transparently measuring strategic reasoning in business simulation platforms.

Requirement 1 *The architecture must support business strategies to be operationalized transparently.*

2.2 Consistent Strategy Operationalization

Transparency alone however does not guarantee a disciplined, consistent strategy operationalization. An important argument for supporting consistency comes from the education domains where business simulations and games are widely used as an educational tool (Wolfe 1993, Faria and Wellington 2004). Business games offer a platform for exploratory learning with little or no guidance, but come with a danger. Several education researchers critique the minimally guided learning approach based on evidence that "it may have negative results when students acquire misconceptions or incomplete or disorganized knowledge" (Kirschner et al. 2006, p. 84). Furthermore, there is hardly any empirical evidence of the cognitive efficacy of business games beyond the self-reports from participants (Anderson and Lawton 2009). Pasin and Giroux (2011) only found four studies that actually measure cognitive-based outcomes for management simulation games, and with mixed results (Pasin and Giroux 2011, p. 1247). Defending business games, Bekebrede et al. (2005) conclude that they "work best when they are embedded into a broader research, learning or intervention process in which a number of complementary methods and activities are used" (Bekebrede et al. 2005, p. 6).

Business games thus offer rich, dynamic experiences but players still require much guidance from experts in order to ensure the pedagogic value of game sessions and fill in the gap between applied theoretic strategies and in-game performance. Following constructivist principles from education theory, Lainema (2009) argues that “the black-box model should be opened. ... This would provide better contexts and assistance to aid the individual in making sense of the environment as it is encountered.” (Lainema 2009, p. 65). This need for learner guidance in business games presents a requirement for supporting consistency of the player’s actions with respect to their chosen strategies.

Requirement 2 *The architecture must support business strategies to be operationalized consistently.*

2.3 Modular Strategy Operationalization

To explore and evaluate business strategies for new business scenarios, the strategy operationalization should also be reconfigurable and extensible at a tactical level. In order to enhance or extend behaviors of simulated actors to reflect existing or new practices, the system users must be given the freedom to change the strategic behaviors and business scenarios completely, rather than being constrained to its original concepts and rule base. A modular approach to strategic decision-making offers several advantages. One advantage of modularity is the strategic flexibility it could offer to the user during a simulation run. Another advantage of modular strategic decision-making is the possibility to add unforeseen phenomena outside the globally shared simulated world, for instance trader cartels that are negotiated peripherally. A third advantage is the opportunity to delegate the simulated actor’s tasks or roles such as sales, purchasing or storage to a team of agents and/or human specialists of arbitrary size. Although Wang et al. (2006) also propose a modular and open reasoning architecture of internal micro-agents, it has some disadvantages: its internal micro-agent architecture is restricted to hierarchical cooperation topologies, the micro-agents are conceptually limited to just representing (partial) reasoning solutions, and the architecture has not been linked to interactive and/or business simulations.

Similarly in agent-based computational economics (ACE) a framework for modular strategic reasoning is not common. For instance, the *Trading Agent Competition for Supply Chain Management* (TAC/SCM) by Sadeh et al. (2003) has been used over the last decade to experiment with and educate about production scheduling, purchasing and sales strategies within a fixed-size computer market scenario. Players enter custom programmed agents into a competition and study their behaviors and performance against various opponents before improving them, but no standard approach is taken other than the Java implementation language. TAC/SCM currently does not support nor allow human interaction, although the TAC community continues to extend the platform to host new setups including agent vs. human competitions (Nelson et al. 2009). Besides its monolithic implementation, the underlying agent platform of TAC/SCM is not very robust nor extensible to new scenarios and setups. This inspired Ketter et al. (2010) to implement their own framework when creating their *Power Trading Agent Competition* (PowerTAC) platform, which aims to extend current ACE methods used specifically to study wholesaler strategies in electrical power markets. However, for the purposes of this paper an extensible business strategy simulation architecture is required that is more topology and domain independent.

To reduce development costs, various solutions have been proposed including the GSCG suite enabling parametrization to support a range of business games for multiple educational settings (van Houten and Verbraeck 2007), the reconfigurability of the *IndustryMasters* game available at industrymasters.com, and a plug and play pathway approach by Tan et al. (2010) for operations management games development. Simulation model reuse and extensibility for supporting new technologies, scenarios and ways of thinking will however remain important concerns to the model users as well as the owners. Modularity in the strategic decision making architecture offers flexibility in design, model reuse, and can thus reduce costs.

Requirement 3 *The architecture must support modular strategic decision making.*

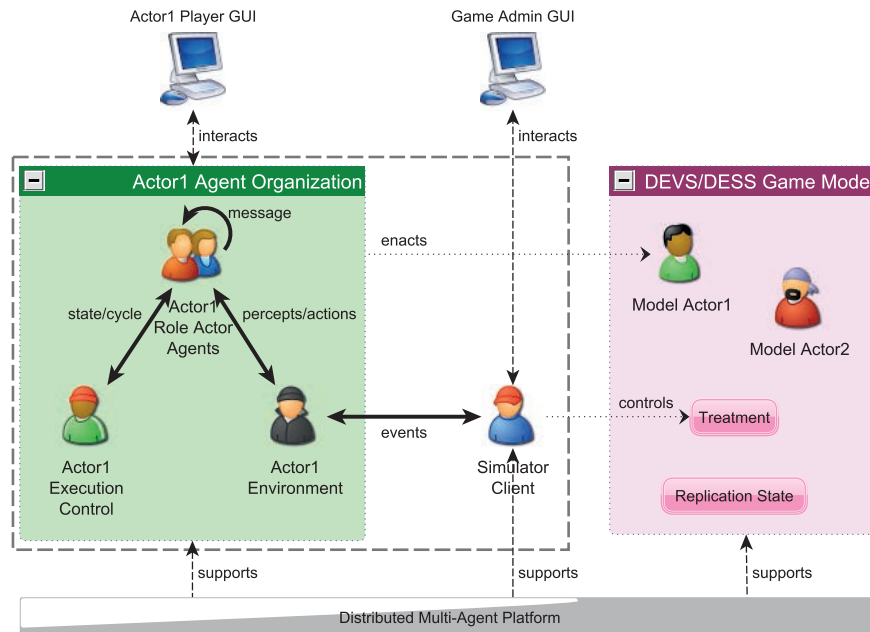


Figure 1: *Modular architecture for distributed interactive business games with local actor reasoning, integrating actor-agent organizations to coordinate a simulated actor's behavior (green area) and a DEVS/DESS game model simulation (purple area) via a distributed multi-agent platform (grey area).*

3 SYSTEM ARCHITECTURE

In order to meet the requirements of transparency, consistency and modularity as specified in the previous section, an architecture for distributed interactive business simulation is proposed that consists of three main components, as depicted in Figure 1. This section briefly discusses how the components separate the concerns of simulation, distribution, and reasoning, as well as their integration within the architecture.

3.1 Components

Successful modularity depends on a clear definition of the interfaces between components. Much in this regard has already been achieved in related research fields of simulation and distributed artificial intelligence in the form of simulation and reasoning formalisms. The concept of a multiple social agents messaging each other while perceiving and acting upon their environment helps to separate concerns for modular reasoning as well as the overall model simulation. The proposed architecture has the following components:

Extensible Agent Middleware Important properties of the agent middleware component, besides being able to support agents, are its robustness, distribution and extensibility. The middleware must be robust to common agent communication and operation failures. Distribution of agents and services across host processors is an essential prerequisite for the intended types of simulation setups. Finally, its extensibility must ensure that the other components can be maintained and modified without causing problems for the distributed setup itself.

Standard Simulator Engine The selected simulator engine must support distributed setups, enable continuous (real-time) simulation, and adhere to standard simulation formalisms specified by Zeigler et al. (2000): the Differential Equations System Specification (DESS) for continuous models, and the Discrete Event System Specification (DEVS) for event-based simulation. These formalisms enable validation of the simulation models on other platforms and simulators in a consistent manner.

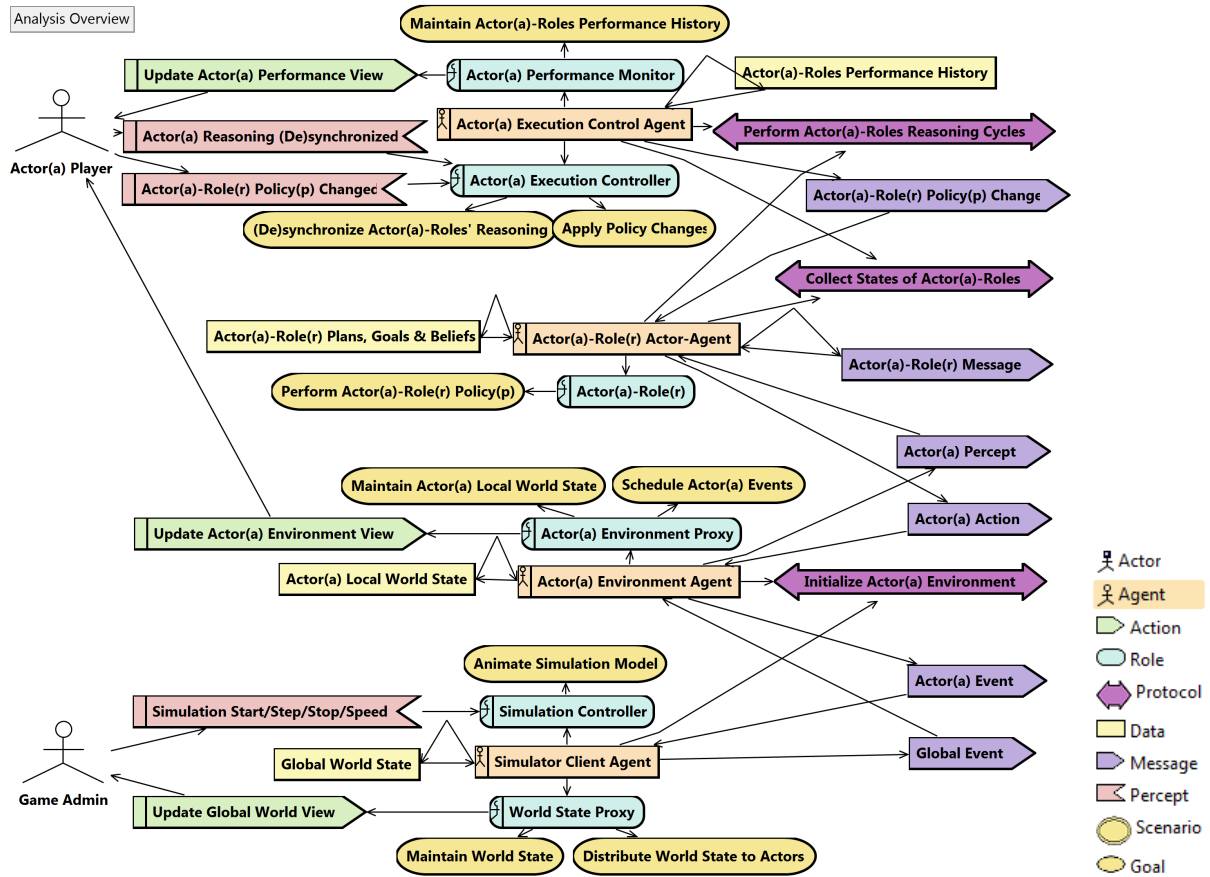


Figure 2: Multi-agent system infrastructure supporting a distributed business simulation where a simulated actor's reasoning is delegated to (possibly several) actor-roles, each performed by an actor-agent.

Modular Reasoning Framework The reasoning framework component localizes reasoning of the simulated actors and supports extensible multi-agent systems that implement the AgentSpeak(L) language by Rao (1996). AgentSpeak(L) is a formal language based on theory of mind and speech act theory where agents reason and communicate based on beliefs, desires and intentions (BDI). Formal reasoning enables validation of the localized reasoning in other agent reasoning frameworks. Enabling introspection and execution control, agents performing various simulated actor roles reason transparently and their behaviors or policies are modifiable and followed consistently.

3.2 Component Integration

To integrate all components, an infrastructure is designed that enables multiple agents to fulfill an extensible set of role policies that together form the strategy of a single simulated actor. This architecture assigns each simulated actor to its own separate reasoning agent organization, such that complexity can be shifted from the central simulation model to the peripheral reasoning engines. By abstracting the actor's behavior to a combination of policies rather than the complete set of individual actions, the overall simulation performance remains the same while actor strategies become clear in the simulation model itself. For instance, a trade actor in the global simulation might distinguish and delegate (and inspect) the sell, store, and purchase roles onto multiple coordinated agents, which then reason locally according to the specified role policies and only the resulting operations are sent back to the global simulation.

The integrating infrastructure's functionality and user-interaction is specified in Figure 2 using the Prometheus multi-agent system design methodology developed by Padgham and Winikoff (2002) and extended by Khallouf and Winikoff (2009). Prometheus applies and extends concepts from Agent-UML and the BDI reasoning framework to provide a goal-oriented description of a multi-agent system's functionalities. More information about the Prometheus methodology and design tool can be found at www.cs.rmit.edu.au/agents/prometheus/.

The infrastructure's analysis overview in Figure 2 shows the system interacting (via the user interfaces depicted in Figure 3) with two user types or Actors: the Game Admin and one Actor(*a*) Player for each simulated strategic actor $a \in A$ who delegates the actor's tactical behaviors $p \in P$ (sales, purchase, inventory, etc.) among their respective actor roles $r \in R_a$. The system itself provides six functionalities or system Roles (cyan) with relevant goals (yellow) which are performed by four types of agents (orange) through system input or Percepts (red) and output or Actions (green). In order to achieve their goals, agents can interact via messages (blue) or message Protocols (purple) and store required data (beige). The infrastructure's roles are as follows:

- Simulation Controller** Performed by the Simulator Client Agent, this system role adjusts the animation of the central simulation model according to perceived simulation *start*, *step*, *stop*, and *speed* commands from the Game Admin.
- World State Proxy** Also performed by the Simulator Client Agent, this system role (i) maintains a global world state to help initialize a strategic actor's environment, and (ii) distributes the current World state events to relevant strategic actor's environment.
- Actor(*a*) Environment Proxy** Performed by an Environment Agent, this system role (i) maintains a local world state for the respective strategic actor *a*, translating events into percepts for the delegated actor-roles, (ii) updates Actor(*a*) Player's environment view accordingly, and (iii) schedules any action taken by the delegated strategic actor-roles.
- Actor(*a*) Execution Controller** Performed by an Execution Control Agent, this system role (i) synchronizes (or desynchronizes) the reasoning cycles of the strategic actor's delegate role agents if commands from Actor(*a*) Player to do so are perceived, and (ii) applies perceived policy changes from Actor(*a*) Player for delegate roles in their respective Actor Agents.
- Actor(*a*) Performance Monitor** Also performed by an Execution Control Agent, this system role (i) maintains the performance history of all the strategic actor's delegated role agents R_a by collecting their current states from the respective Actor Agents, and (ii) updates the Actor(*a*) Player's performance view for the strategic actor based on its performance history emerging from the delegate agents' selected policy combination.
- Actor(*a*)-Role(*r*)** Performed by an Actor Agent, this system role performs (tactical/operational) policy *p* for actor role *r* as delegated by the (strategic) Actor(*a*) Player.

4 SYSTEM IMPLEMENTATION

To explore the feasibility of the architecture design, an implementation was made with a selection of open-source, Java-based (thus platform-independent) components: the AgentScape agent middleware, the DSOL based GSCG distributed business simulation suite, and the Jason agent reasoning framework.

4.1 AgentScape Agent Middleware

The open-source AgentScape middleware (Brazier et al. 2002, Overeinder and Brazier 2006) is selected for its robustness, scalability and extensibility. The middleware, used in large production environments as well as many research projects, is under constant development and was recently rebuilt to support more extensibility. More information about the AgentScape middleware can be found at www.agentscape.org.

4.2 DSOL and the Global Supply Chain Games Suite

The open-source *Distributed Simulation Object Library* (DSOL) by Jacobs et al. (2002), Jacobs (2005) implements and combines the DEVS and DESS simulation formalisms specified by Zeigler et al. (2000). DSOL replicates Java-based DEVS and/or DESS model simulations based on specified treatment parameters such as random seeds, warm-up time, etc. The *Distributor Game*, part of the *Global Supply Chain Games* (GSCG) suite (van Houten and Jacobs 2004, van Houten et al. 2005) mentioned in Section 1, is implemented with a DSOL-based supply chain training library (Verbraeck and van Houten 2005). More details about simulation with DSOL can be found at simulation.tudelft.nl, and about the GSCG suite at gscg.org.

4.3 Jason Multi-Agent System and Reasoning Framework

Jason is an open-source framework by Bordini et al. (2007) that implements the AgentSpeak(L) beliefs-desires-intentions (BDI) reasoning language by Rao (1996). In the Jason framework, multi-agent systems are composed of an environment, a group of agents, and possibly an execution controller to synchronize the agents' reasoning steps and inspect the agents' current states if so desired. Jason is also compatible with the Prometheus methodology (Padgham and Winikoff 2002, Khallouf and Winikoff 2009). Jason's agent, environment and execution control classes for introspection and policy modification can easily be extended to fit specific needs. More information about the Jason framework can be found at jason.sf.net.

4.4 Jason-DSOL Integration Using AgentScape

A new infrastructure of tier-agents was implemented on the AgentScape middleware, similar to Jason's standard tier-agents implemented for the Jade agent middleware (Bellifemine et al. 2001). Unlike Jade agents, AgentScape agents are completely mobile, enabling Jason agents to reason for the simulated actors in localized setups whilst distributed dynamically across remote clients. Performance of this Java implementation will be evaluated and compared in future work.

5 USE CASE

To illustrate the benefits of the proposed simulation architecture, this paper shows how it could support redesigning the Distributor Game, part of the business games suite by van Houten et al. (2005), which focuses on supply chain (or demand network) logistics. Aimed at educating the players in the field of business strategy, the current game-play in the suite's business games such as the *Distributor Game* typically revolves around operationalizing business strategy. After choosing their business strategies, players are assumed to execute them consistently, performing whichever operations they see fit the chosen strategy. Since the *Distributor Game*'s evaluation by van Houten (2007), this paper's authors confirmed through observation during follow-up sessions that, as students try to operationalize their strategy, it is hard for educators (a) to determine whether students follow their intended strategy consistently, (b) to determine exactly when they possibly changed it, and (c) to modify the challenges (presented for instance by demand and supply dynamics) that currently best fit the entire group's learning process. In short, the Distributor Game does not meet the requirements discussed in Section 2.

To meet with the requirements of transparent, consistent and modular strategic decision-making by the simulated actors, the game can be redesigned with the proposed architecture as follows.

1. To improve transparency, the simulation model is formalized as a DESS-in-DEVS system specification suited for the simulation service. Behavioral rules or policies governing simulated actors are formalized in the Prolog-based Jason reasoning language, ready for validation by experts on respective management theory. The educator is informed of all policy changes the players make during a simulation. This will show how their strategic decisions (and rationales) contributed to their overall performance in the simulation. Educators can now base their feedback on accurate information about how players operationalized their strategy.

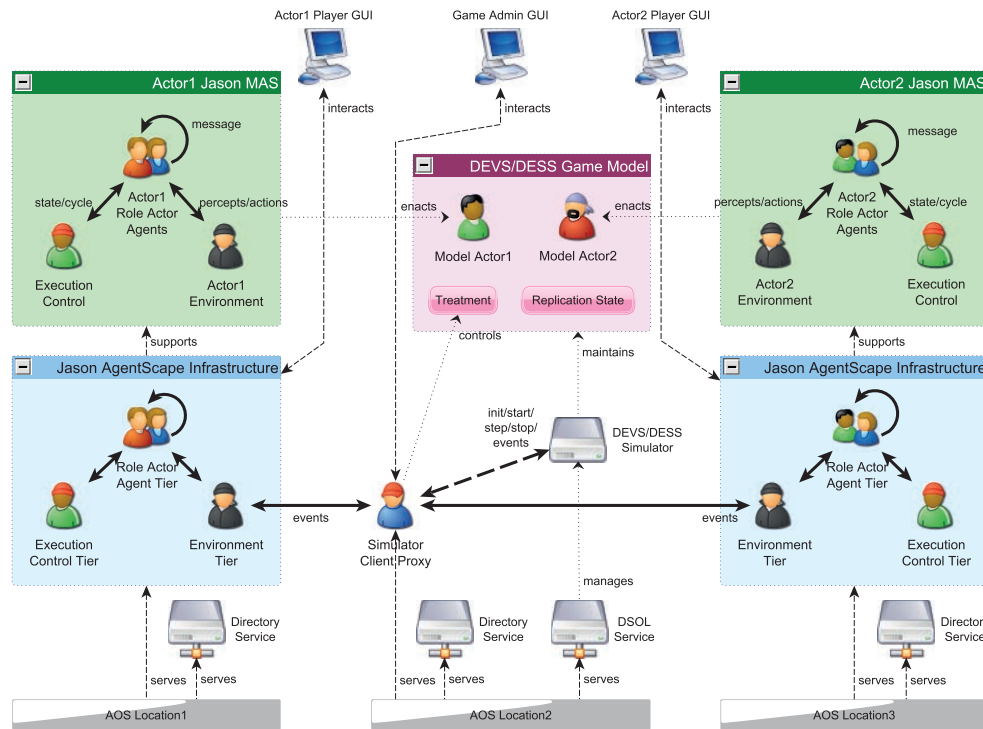


Figure 3: *Implementation of the agent architecture for interactive business games with localized actor reasoning. Agent organizations (green areas) composing a single actor's strategy are implemented in Jason and supported by mobile agents (blue areas). The game model simulation (purple area) is supported separately by a simulation service and client proxy agent. Agents and services run on the agent middleware.*

2. To improve consistency, strategies are delegated and performed (operationalized) by tactical Jason actor-role agents which execute any formalized strategy without mistake (although players' strategic choices for particular tactics may still be inappropriate). In the *Distributor Game* scenario, the demand and supply behaviors are controlled by the educator as game facilitator, whereas the distributor actor behaviors are controlled by the learners through their actor's delegate roles. Strategic mistakes can now be measured and applied by educators to provide appropriate feedback to help players improve and learn to make their policy changes systemic and consistent with their overall business strategy.
3. To improve modularity, actor strategies are decomposed and into (sub)strategies or tactics for relevant fields of expertise (purchase, sales, inventory, etc.). Each tactic is delegated to a separate tactical Jason actor-agent within an agent organization that is making decisions for one particular actor. Adding, removing or replacing these delegate agents reflects a change in overall actor strategy, for instance to mitigate or recover from some supply chain risk event observed by the player.

6 CONCLUSION

This paper proposes an architecture for distributed interactive business simulations to answer the main question of how to design a system that facilitates transparency, consistency and modularity of strategic decision making by simulated actors in interactive business simulations. The system's first requirement, that it supports business strategies to be operationalized transparently, is achieved through a formal reasoning language to specify how tactical policies operationalize parts of actor strategies, for instance purchase, sales, inventory, etc. The system's second requirement, that it supports business strategies to be operationalized consistently, is achieved through the ability to monitor and inspect the executed policies and their interactions.

The third system requirement, that it supports modularity in strategic decision making, is achieved with a multi-agent infrastructure enabling on-the-fly modification and (re)combination of actor (sub)strategies.

In future work the authors aim to verify that the transparency and consistency offered by this architecture actually contributes to and simplifies the validation of the business simulation itself and thus improves the players' mental models. Literature suggests that improving players' mental models will consequently improve their business performance. Furthermore, the modularity of the architecture is expected to contribute to the scalability of the business simulations that implement it. The effects of localization of reasoning on scalability deserve further investigation, as do performance effects of different configurations, for example with other simulation engines, reasoners, or individual actor implementations.

Transparency and consistency have been argued to be absolutely essential for business games to succeed as effective learning tools. Misconceptions due to lack of guidance can be prevented by providing correct insights about what causes might have led to the observed effects. That is, the game should provide a clear picture of how business strategies (and mistakes) actually led to the simulated actors' performance. The presented architecture facilitates precisely these aspects and thus enhances business game effectiveness.

ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their comments. Furthermore, financial support from the Delft University of Technology and the Next Generation Infrastructures Foundation is also gratefully acknowledged for this research.

REFERENCES

- Anderson, P. H., and L. Lawton. 2009, April. "Business Simulations and Cognitive Learning". *Simulation & Gaming* 40 (2): 193–216.
- Bekebrede, G., I. S. Mayer, S. P. A. van Houten, R. T. H. Chin, and A. Verbraeck. 2005. "How serious are serious games? Some lessons from Infra-games". In *Proceedings DiGRA 2005 Conference Changing Views: Worlds in Play*, edited by S. de Castell and J. Jensen, 1–7: DiGRA.
- Bellifemine, F., A. Poggi, and G. Rimassa. 2001, 7-9 July. "Developing Multi-agent Systems with JADE". In *ATAL'00: Proceedings 7th Int'l Workshop on Intelligent Agents: Agent Theories, Architectures and Languages*, edited by C. Castelfranchi and Y. Lespérance, Volume 1986 of *LNCS*, 42–47: Springer-Verlag.
- Bordini, R. H., J. F. Hübner, and M. Wooldridge. 2007, October. *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons.
- Brazier, F. M. T., D. G. A. Mobach, B. J. Overeinder, S. van Splunter, M. van Steen, and N. J. E. Wijngaards. 2002, 27-31 May. "AgentScape: Middleware, Resource Management, and Services". In *SANE'02: Proceedings 3rd Int'l Conference on System Administration and Network Engineering*, 403–404.
- Corsi, T. M., S. Boyson, A. Verbraeck, S. P. A. van Houten, C. Han, and J. R. MacDonald. 2006. "The Real-Time Global Supply Chain Game: New Educational Tool for Developing Supply Chain Management Professionals". *Transportation Journal* 45 (3): 61–73.
- Faria, A. J., D. Hutchinson, W. J. Wellington, and S. Gold. 2009, August. "Developments in Business Gaming: A Review of the Past 40 Years". *Simulation & Gaming* 40 (4): 464–487.
- Faria, A. J., and W. J. Wellington. 2004, June. "A Survey of Simulation Game Users, Former-Users, and Never-Users". *Simulation & Gaming* 35 (2): 178–207.
- Gary, M. S., and R. E. Wood. 2010, June. "Mental models, decision rules, and performance heterogeneity". *Strategic Management J.* 32 (6): 569–594.
- Geurts, J. L., R. D. Duke, and P. A. Vermeulen. 2007, December. "Policy Gaming for Strategy and Change". *Long Range Planning* 40 (6): 535–558.

- Gosen, J., and J. Washbush. 2004, June. "A Review of Scholarship on Assessing Experiential Learning Effectiveness". *Simulation & Gaming* 35 (2): 270–293.
- Jacobs, P. H. M. 2005, 15 November. *The DSOL simulation suite*. Ph. D. thesis, Delft University of Technology, Delft, The Netherlands.
- Jacobs, P. H. M., N. A. Lang, and A. Verbraeck. 2002. "D-SOL: a distributed Java based discrete event simulation architecture". In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 793–800. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Ketter, W., J. Collins, and C. Block. 2010, December. "Smart Grid Economics: Policy Guidance through Competitive Simulation". Technical Report ERS-2010-043-LIS, Erasmus Res. Inst. of Mgmt.
- Khallouf, J., and M. Winikoff. 2009. "The goal-oriented design of agent systems: A refinement of Prometheus and its evaluation". *Int'l J. of Agent-Oriented Software Engineering* 3 (1): 88–112.
- Kirschner, P. A., J. Sweller, and R. E. Clark. 2006, June. "Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching". *Educational Psychologist* 41 (2): 75–86.
- Lainema, T. 2009, February. "Perspective Making – Constructivism as a Meaning-Making Structure for Simulation Gaming". *Simulation & Gaming* 40 (1): 48–67.
- Lainema, T., and O.-P. Hilmola. 2005. "Learning More, Better and Faster? Computer-Based Simulation and Gaming of Production and Operations". *Int'l J. of Business Performance Management* 7 (1): 34–59.
- Nelson, A., D. Nyabuti, J. Collins, W. Ketter, and M. Gini. 2009, 13 July. "Toward Human-Agent Competition in TAC SCM". In *TADA-09: IJCAI Workshop on Trading Agent Design and Analysis*, edited by E. H. Gerding: Association for Trading Agent Research / AAAI.
- Overeinder, B. J., and F. M. T. Brazier. 2006. "Scalable Middleware Environment for Agent-Based Internet Applications". In *Applied Parallel Computing*, Volume 3732 of *LNCS*, 675–679. Berlin: Springer-Verlag.
- Padgham, L., and M. Winikoff. 2002, 15 July. "Prometheus: A Methodology for Developing Intelligent Agents". In *AOSE 2002: 3rd Int'l Workshop on Agent-Oriented Software Engineering*, edited by F. Giunchiglia, J. Odell, and G. Weiss, Volume 2585 of *LNCS*, 174–185: Springer-Verlag.
- Pasin, F., and H. Giroux. 2011, August. "The impact of a simulation game on operations management education". *Computers & Education* 57 (1): 1240–1254.
- Rao, A. S. 1996. "AgentSpeak(L): BDI agents speak out in a logical computable language". In *MAAMAW'96: Proceedings 7th Eur. workshop on Modelling Autonomous Agents in a Multi-Agent World: Agents breaking away*, 42–55: Springer-Verlag.
- Sadeh, N., R. Arunachalam, J. Eriksson, N. Finne, and S. Janson. 2003. "TAC-03: A Supply-Chain Trading Competition". *AI Magazine* 24 (1): 92–94.
- Summers, G. J. 2004, June. "Today's Business Simulation Industry". *Simulation & Gaming* 35 (2): 208–241.
- Tan, K. H., Y. K. Tse, and P. L. Chung. 2010. "A plug and play pathway approach for operations management games development". *Computers & Education* 55 (1): 109–117.
- Thavikulwat, P., and S. Pillutla. 2010. "A constructivist approach to designing business simulations for strategic management". *Simulation & Gaming* 41 (2): 203–230.
- van Houten, S. P. A. 2007, 6 November. *A suite for developing and using business games: supporting supply chain business games in a distributed context*. Ph. D. thesis, Delft University of Technology, Delft, The Netherlands.
- van Houten, S. P. A., and P. H. M. Jacobs. 2004. "An architecture for distributed simulation games". In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 2081–2086. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- van Houten, S. P. A., and A. Verbraeck. 2007. "Supporting parametrization of business games for multiple educational settings". In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G.

- Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- van Houten, S. P. A., A. Verbraeck, S. Boyson, and T. Corsi. 2005. "Training for today's supply chains: An introduction to the Distributor Game". In *Proceedings of the 2005 Winter Simulation conference*, edited by M. Kuhl, N. Steiger, F. Armstrong, and J. Joines, 2338–2345. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Verbraeck, A., and S. P. A. van Houten. 2005. "From simulation to gaming: an object-oriented supply chain training library". In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. Kuhl, N. Steiger, F. Armstrong, and J. Joines, 2346–2354. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Wang, M., M. Nowostawski, and M. Purvis. 2006, 26 July. "Declarative Agent Programming Support for a FIPA-Compliant Agent Platform". In *ProMAS 2005: 3rd Int'l Workshop on Programming Multi-Agent Systems—Revised and Invited Papers*, edited by R. Bordini, M. Dastani, J. Dix, and A. El Fallah Seghrouchni, Number 3862 in LNCS, 252–266. Utrecht, The Netherlands: Springer-Verlag.
- Wolfe, J. 1993, December. "A History of Business Teaching Games in English-Speaking and Post-Socialist Countries: The Origination and Diffusion of a Management Education and Development Technology". *Simulation & Gaming* 24 (4): 446–463.
- Zeigler, B. P., H. Praehofer, and T. G. Kim. 2000. *Theory of Modeling and Simulation—Integrating Discrete Event and Continuous Complex Dynamic Systems*. 2nd ed. San Diego, CA: Academic Press, Inc.

AUTHOR BIOGRAPHIES

RICK VAN KREVELEN is a researcher and PhD candidate at the Systems Engineering Group of the Delft University of Technology, The Netherlands. He holds a Master's in Artificial Intelligence from the VU University Amsterdam. Drs. Van Krevelen researches Agent Technology for Business Games and Simulations. His e-mail address is d.w.f.vankrevelen@tudelft.nl.

MARTIJN WARNIER is an Assistant Professor at the Systems Engineering Group of the Delft University of Technology, The Netherlands. He holds a PhD in Computer Science from the Radboud University Nijmegen. Dr. Warnier's research concerns the design of large scale multi-agent based distributed systems. His e-mail address is m.e.warnier@tudelft.nl.

FRANCES BRAZIER holds the Engineering Systems Foundations Chair at the Delft University of Technology, where she chairs the Systems Engineering Group of the Faculty of Technology, Policy and Management. She is an expert in the design of large scale participatory systems: systems characterized by distribution of governance, autonomy, intelligence, engagement, and communication. Her e-mail address is f.m.brazier@tudelft.nl.

ALEXANDER VERBRAECK is a Full Professor of Systems and Simulation at the Systems Engineering Group of the Delft University of Technology, The Netherlands, and a part-time Research Professor at the R. H. Smith School of Business of the University of Maryland. Professor Verbraeck is a specialist in discrete event simulation for real-time control of complex transportation systems and for modeling business systems. His e-mail address is a.verbraeck@tudelft.nl.

THOMAS CORSI is the Michelle Smith Professor of Logistics at the Robert H. Smith School of Business of the University of Maryland. Professor Corsi is the Director of the Supply Chain Management Center. His e-mail address is tcorsi@rhsmith.umd.edu.