Architecture Design for System-Integrated Business Simulation Games

Bastian Kurbjuhn, Klaus Turowski

Very Large Business Applications Lab (FIN-ITI-VLBA)
Otto von Guericke University Magdeburg
Universitätsplatz 2
39106 Magdeburg, ST, Germany
Bastian.Kurbjuhn@ovgu.de
Klaus.Turowski@ovgu.de

Abstract: Enterprise Resource Planning (ERP) systems are a substantial part of modern business. For this reason it is not astonishing that their complexity grows with assigned requirements. Therefore, it is vital and sustainable to invest into recruitment of new ERP experts. System-integrated (business) simulation games form a new educational approach adapting theoretical concepts on a practical framework. In this paper, an architectural design proposition for such systems is presented. First considerations revealed that several technical and organisational challenges have to be solved: e.g. the integration into the system landscape of a software provider, the communication through interfaces or the pedagogical objective. All those issues may influence one or multiple components or even the entire architecture itself.

Keywords: System-Integrated Simulation Games, Architecture

1 Introduction

Enterprise Resource Planning (ERP) systems are business applications consisting of standardised, enterprise-wide *"software packages"*. Their comprehensive functionality provides business process integration across the company [Da98]. Following Davenport, business processes are *"a structured, measured set of activities designed to pr*oduce a *specific output for a particular customer or market"* [Da93]. One effect of globalisation is that business complexity[1] increases, therefore requirements for ERP systems still arise. Consequently, the use of such systems with increasing functionality requires recruitment of experts or qualified employees with interdisciplinary knowledge.

Being faced with that issue, business schools have implemented several pedagogical strategies [Se07] (e.g. case studies or team teaching) and have created special learning

[1] Modern supply chain concepts across the whole value chain like just in time or even just in sequence logistics have to consider intercompany processes as well.

and teaching environments [VBM01]. However, simulations – used by enterprises, consultancy companies and business schools [Fa98; Mo01] – are one of the most preferred tools. Due to their nature of reducing complexity of a particular problem without distorting reality [Se08] and of combining theory and practise, they are especially suitable for classroom teaching.

Classic, stand-alone running business simulation games – like TOPSIM General Management II[2] or iDECOR[3] – focus on organisational, managerial matters and on enterprise leadership (decision making process) [RH11, p. 265]. Multiple companies compete with each other [Se08, p. 5]. But this type of simulation games do not cover application-related topics like system configuration or – on a higher business level – system customisation which are crucial factors for efficient operation of business applications. In order to combine merits of business simulations with this claim for system handling, the approach of so-called system-integrated business simulation games seems to be very promising. Such games consist of a business simulation which is connected to a business application.

In order to design and set up an ERP-integrated simulation game, several technical and organisational challenges have to be considered and solved. The focus of this contribution lies in the presentation of a first draft for system-integrated business simulation architecture.

The paper is structured as follows: First, a summary about simulation games and relations to existing work are compiled. After that related work section, the paper outlines a draft for a high-level technical architecture which is grounded on a previous requirements analysis. Finally in the conclusion part, some challenges towards implementation and operation of such games are identified.

2 Related Work

Simulation games belong to the Serious Games family. That term covers all games with an *"explicit and carefully thought*-out educational purpose and are not intended to be *played primarily for amusement"* [Ab70, p. 6].

Geilhardt and Mühlbradt compare a simulation game with a constructed situation, whose participants' behaviour has to be inherent with predefined rules of surrounding model [GM95, p.11]. The game leader observes the situation and analyses their decisions. Kritz identifies three components for assembling a simulation game [Kr01, p. 52]:

- **Simulation** (of closed system) describes technical implementation of given model.
- **Game** represents the set of rules in order to structure and define the processes.
- **Role** stands for the human functions the participants take in dynamic systems.

[2] TOPSIM General Management II is copyrighted by TATA Interactive Systems.
http://www.topsim.com/de/planspiele/general_management_ii/
[3] iDECOR is based on DECOR simulation and under copyright by H. H. Schrader.
http://www.idecor.de/produkt/index.shtm

With the extension of its application field due to combination of various disciplines (strategic/tactical military games, operations research and computer science) in 1950s, serious games – especially simulations – became more and more popular to civil sector. In particular, companies were interested in tools for developing managerial skills along decision making process [RH11, pp. 264-265]. Several later developed games (e.g. "IN-TOP" in 1964, "BEER Game" in 1985) represent complete (production) companies with the focus on collaboration and process management.

Although serious games are subdivided into multiple other application fields (e.g. health and corporate sector), this paper hereafter deals with business game domain.

2.1 Classic Business Simulation Games

Classic simulation games are normally structured as follows (e.g. TOPSIM and iDECOR): X companies, represented by a team of multiple participants, compete against each other with a maximum on Y products on Z markets [RS01, p. 1]. Starting with same conditions, the participants have to develop and discuss strategies in order to reach primary targets like (long term) solvency and viability. Secondary aims might be raising market share or being leader in a special discipline (e. g. selling high-tech products or expanding into new markets).

A classic business simulation normally begins with an introduction where the participants are separated into teams. The game manual represents the bundle of rules that delimits the horizon of decision; e. g. a team cannot buy the fourth machine when only three are allowed. During a game round, the candidates meet within the team to organise themselves and discuss strategies, so that decisions can be derived. When decisions are made, they are sent to the game leader. After having collected all decision data, the game leader starts the simulation in order to process new values for the following business period. In the end, the game leader presents overall performance and declares the winning team.

2.2 System-Integrated Business Simulation Games

The particular difference towards classic business simulation games is that those simulation games are system-dependant. Especially system-integrated business simulation games predominantly scope on ERP systems.

Whilst classic simulation games predominantly focus on managerial and organisational matters, system-integrated simulation games also consider technical issues like system customizing and handling. In contrast to classic simulation games, where the participants get business reports from the game leader every period, important key performance indicators and reports have to be pulled from the ERP system autonomously. Due to this fact, such games are not only interesting for students in business administration, they may target students of technical affairs (e. g. business information systems), too. Otherwise, this sort of simulation games can even be interesting for companies to simulate the effectiveness of planned innovations or modifications of their business strategy.

One of those first developed system-integrated business simulation games is called ERPsim[4]. That game is implemented as a Java web application which is coupled to the ERP system SAP ERP 6.0 [Le07]. Léger et al. describe it as an innovating training environment *"to address the challenges in learning to use and understand ERP systems"* [Le11, p. 38].

3 Architecture Design

Following the integrated simulation game approach, their design and implementation do not mark a trivial process, because with the connection of an entire simulation engine to a reality-grounded information system, several technical and organisational challenges have to be identified and solved.

This chapter is dedicated to propose a theoretical architecture draft. This solution identifies necessary components and investigates in possible connection alternatives. At the beginning, a requirement analysis is performed outlining the objectives for the architecture.

3.1 Requirement Analysis and Objectives

A requirement analysis ensures that requirements, being addressed towards a software system, are correctly and completely formulated, meet conformity regulations, are consistent and feasible [Du03].

The proposed architecture for integrated simulation games follows different objectives than ERPsim. That game is mainly designed for academic sector and considers exclusively educational purposes. That approach finds expression in a fixed starting scenario, a predefined menu of usable transactions and customized reports.

In this paper, the architectural proposition shall meet other requirements which are listed below.

Functional and Qualitative Requirements

As mentioned in the prior section, a technical implementation of a simulation game shall ensure maximum flexibility. The ERP system may only be reduced or simplified where necessary or where it is reasonable according to simulation. That implies:

- The simulation of different scenarios shall be supported ("manoeuvres simulation").
- Interaction and planning steps have to be done in ERP system environment. Only the administrator or game leader controls the simulator.
- Communication (in- and output of data) between simulation component and ERP system will be accomplished by the use of pre-defined, preferably standardised inter-

[4] ERPsim is developed by ERPsim Lab of HEC Montréal. http://erpsim.hec.ca/en/about

faces (e.g. by calling SAP BAPI[5] functions or using industry standards like EDIFACT[6] or other B2B[7] interfaces). This intends seeing a simulation game as a platform which is not only limited to one ERP product or provider.

Administrative, System-Related Requirements

Regardless of its functionality, efficient operation of a complex application system (if the entire simulation game with its host system is considered) is an important factor of success for organisations. Otherwise, target group expansion towards such an integrated game will not succeed.

Therefore, integration and operation might not bind many additional resources or cause too many exceptions in relation to "normal" operation of a system landscape.

- The system should be designed for long-term operation. Shutting the system down every day or restart it for every started round has to be avoided.
- Maintaining the system must not be elaborative. This aspect intends a special configuration so that unnecessary background tasks are disabled (e. g. backup jobs).
- The entire simulation game (host system and simulation engine) must run in a stable environment and must not be interrupted from external sources or internal programming/configuration errors.

3.2 Theoretical Draft

Before dealing with the design of the high-level architecture proposition, we have to take the coupling into account.

Possible Coupling Configurations

Figure 1 identifies three different coupling alternatives in order to establish communication between simulation component and ERP system. These two parts constitute the integrated simulation game.

The integrated approach (a) aims at constructing the entire simulation logic within the ERP system. In contrary, the distributed variant (c) seeks in outsourcing the simulation component to another application server (e.g. as a web application). The mixed mode provides some parts of the simulation which are implemented in the ERP system; other parts may not depend on ERP runtime but require installation on the same machine.

[5] Business Application Programming Interfaces developed by SAP provide external access to business objects and processes in SAP ERP R/3 and newer.
[6] Electronic Data Interchange For Administration, Commerce and Transport
[7] Business to Business

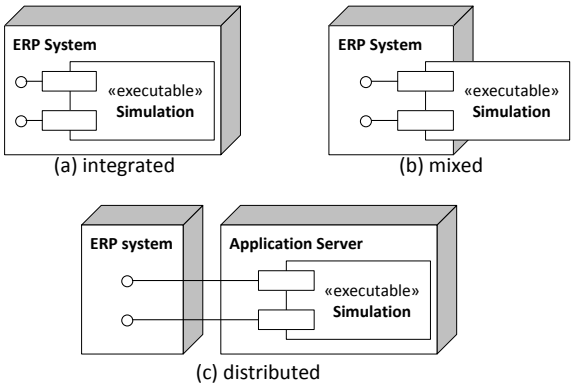

Figure 1: Coupling variants (UML distribution diagram)

The integrated solution (a) will be an appropriate approach for a simple simulation model. The simulation program code has to be written in the language of its host application. The advantage lies in execution of the simulation within the runtime environment of the ERP system so that data processing can be done without the use of external interfaces. But this tight coupling abandons portability.

The mixed solution (b) – a strong coupling – is difficult to maintain and difficult to integrate into an existing system landscape. This solution might be appropriate for a simulation engine with a few components.

A distributed system design (c) is beneficial for a complex simulation model and meets requirements for complex system landscapes. In relation to system stability, a failure of one component (ERP system or simulation engine) will not have a severe impact towards its opponent. In addition to that, such a loosely-coupled system even fulfils portability requirements. In this context, due to its independency, the simulation component turns into the role of a simulation platform dealing with different ERP systems.

High-Level Architecture Framework

Having the distributed connection alternative in mind (refer to Figure 1 (c)), the simulation component of a system-integrated simulation game is made up of three parts:

- **Simulation model** – contains the set of rules and logic,
- **Simulation engine** – performs communication between ERP system and simulation model and
- **Simulation platform** – defines and provides communication channels (standard interfaces).

Figure 2 presents an architectonic design proposition for system-integrated business simulation games on a high, theoretical level: Referring to Davenport's definition of ERP systems[8], the visualised business modules correspond to Davenport's term of

[8] Refer to chapter 1 (Introduction) and [Da98].

"software packages". Either they or ERP system core provide system-specific or standardised interfaces for communicational issues and data interchange.

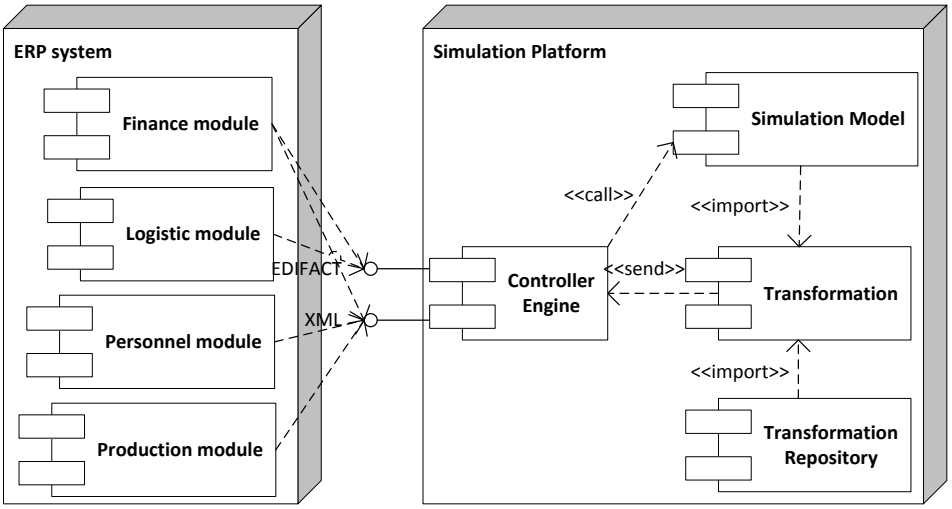

Figure 2: Architecture model (UML component diagram)

When simulation model finished evaluation of decisions and generated output (e.g. customer demands, analysis and evaluation of financial statements), a transformation component has to be fed with the result data of the simulation model because ERP systems operate on a more granular level and therefore need additional information in order to perform a transaction and generate documents. The main task of the transformation component is to convert aggregated data into transactional statements. To enrich derived transactions with additional information, a data repository is used as a source pool to complete transactional statements (e.g. master data like name of employees, description of customers etc.). After the values of simulation model are completely transformed into transactions, they are returned to the controller. It is responsible for external and internal communication.

3.3 Communication Between ERP System and Simulator

Figure 3 visualises the communication flow between the ERP system and the various components of the simulation engine. In order to describe each step, let us assume that a client has decided in period 1 to hire three new employees for next period.

After period 1, the game leader either enables the simulation or simulator is triggered automatically. The simulation controller then calls the simulation model to fetch all relevant decision data. During data processing, the model has calculated that only one vacancy will be occupied. After the model has quitted work, the controller passes calculations to the transformation engine. The task of the transformation engine is to prepare a transformation according to a SAP ERP BAPI specification. With the help of the transformation repository – acting as data pool provider – it may collect missing data for the

personal file (e.g. `name: Merlère, prename: Guy de, birthday: 12.03.1990` etc.). When all essential data is collected from the data pool, the ERP system will be fed with this data set using an appropriate BAPI of the Human Capital Management module.

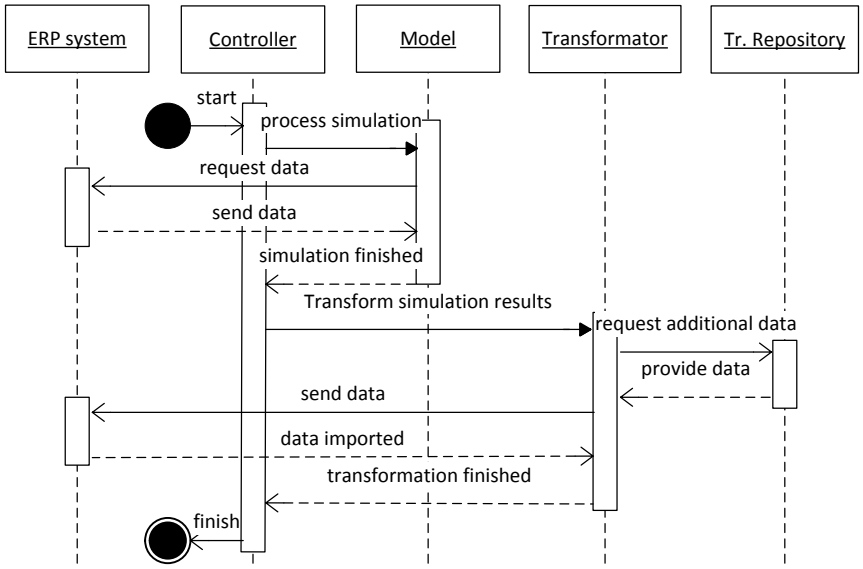

Figure 3: Communication flow between ERP system and simulator (UML sequence diagram)

4 Summary and Outlook

It is found out that the design of a system-integrated simulation game is not a trivial process. On the technical level, the architect has to answer two questions:

1. How should the game be coupled with the host system?
2. Which software components and interfaces are required for the architecture framework?

The answer to the first question depends on the size and explicit scope of simulation model and on the operational requirements. But the distributed approach offers high potential due to its platform-supporting design (portability).

In order to elaborate and answer the second question, it is found out that many distributed business applications consisting of an application and database system process data on transactional level. In order to establish a communication between simulation and the

business application, it is vital to construct a transformation area in order to disaggregate summarized information into transactions.

Looking forward to integrating such systems to a system landscape implies special requirements. It is appreciated that such systems indulge a special configuration (switching off background/scheduled tasks), otherwise problems might occur during system operation. Even for operation, it is vital to conceptualise an operations model describing preparing and finalising processes because the system reset and preparation of a new round might be a challenging task on a combined system.

In the context of the cloud computing paradigm, simulation game architectures are confronted with accessibility and scalability issues and have to be adapted for web service communication for the chosen service and deployment models. All in all, the design of an integrated simulation game depends on its technical environment.

In addition to that, pedagogical issues also influence the design and scope of the game. Shall the participants compete with each other (as concurrent enterprises) or shall the game be adjusted along the value-added chain through a company so that a team corresponds to a single department? The curriculum design marks a challenging task and may affect the simulation model.

References

[Ab70] Abt, C. C.: Serious Games. Viking Press, New York, 1970.

[Da93] Davenport, T. H.: Process Innovation – Reengineering Work through Information Technology. Harvard Business School Press, Boston, Massachusetts, USA, 1993.

[Da98] Davenport, T. H.: Putting the Enterprise into the Enterprise System. In: Harward Business Review, vol. 75, issue 4 (July/August), Harvard Business School Publishing, Boston, Massachusetts, USA, 1998; pp. 121-131.

[Du03] Dumke, R.: Software Engineering. 4th edition, vieweg, Wiesbaden, 2003

[Fa98] Faria, A. J.: Business Simulation Games: Current Usage Levels – An Update. In: Simulation & Gaming 29 (3), 1998; pp. 295-309.

[GM95] Geilhardt, T.; Mühlbradt, T.: Planspiele im Personal- und Organisationsmanagement. Hogrefe, Göttingen, 1995.

[Kr01] Kritz, W.-C.: Die Planspielmethode als Lernumgebung. In: Mandl, H.; Keller, C.; Reiserer, M.; Geier, G. (eds.): Planspiele im Internet – Konzepte und Praxisbeispiele für den Einsatz in Aus- und Weiterbildung, vol. 26, wbv, Bielefeld, 2001; pp. 78-95.

[Le07] Léger, P.-M.: Using a Simulation Game Approach to Teach Enterprise Resource Planning Concepts. In: Journal of Information Systems Education 17 (4), 2007; pp. 441-447.

[Le11] Léger, P.-M., et al.: Business Simulation Training in Information Technology Education: Guidelines for New Apporaches in IT Training. In: Journal of Information Technology Education 10, 2011; pp. 37-50.

[MHR01] Moratis, L.; Hoff, J.; Reul, B.: A Dual Challenge Facing Management Education: Simulation-Based Learning and Learning about CSR. In: The Journal of Management Development 25 (3/4), 2001; pp. 213-231.

[RH11] Riedel, J. C. K. H.; Hauge, J. B.: Evaluation of Simulation Games for Teaching Production (Engineering). In Cruz-Cunha, M. M.; Carvalho, V. H.; Tavares, P. (eds.): Computer Games as Educational and Management Tools – Uses and Approaches. Information Science Reference, Hershey, PA, USA, 2011; pp. 263–279.

[RS01] Rautenstrauch, C.; Schrader H. H.: iDECOR – Das Spielerhandbuch zum Unternehmensplanspiel. Magdeburg, 2001, ISBN: 3-929757-43-5.

[Se07] Seethamraju, R.: Process Orientation to Business Students – Enabling Role of Enterprise Systems in Curriculum. In: Proceedings of the 18th Australasian Conference on Information Systems, Toowomba, Queensland, Australia, 2007.

[Se08] Seethamraju, R.: Enhancing Student Learning through ERP Business Simulation Game. In: Proceedings of the AIS SIG-ED IAIM 2008 Conference, 2008.

[VBM01] Van Baalen, P. J.; Moratis, L. T.: Management Education in the Network Economy – Its Context, Content and Organization. Kluwer Academic Publishers, Dordrecht, Netherlands, 2001.