Interaction of Simulation Tools with ERP Systems
Concept and Practical Implementation

Arne Koors and Bernd Page
Department of Informatics, University of Hamburg, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany

Keywords: Discrete Event Simulation, ERP Systems, ERP System Utilisation, System Interaction.

Abstract: The interaction approach introduced in this paper is aiming at coupling a full-fledged discrete event simulator and an operational ERP system in a fully integrated manner. Here, the simulator is representing the complete operative environment of the ERP system, substituting its daily business input. For this, the company-specific utilisation of the ERP system has to be modelled in the simulator. In order to execute its model and process concrete ERP functionality, the simulator is accessing the ERP system via software interfaces, using it like a large subject-specific software library. Thus, the simulator is effectively carrying out a complete remote control of the ERP system, in the sense of software automation. Amongst others, the simulator is inducing arrival and booking events in the ERP system, is continuously triggering internal ERP system processes and is processing the results of revised ERP planning by arranging future events in the ERP system. Altogether, the simulator and the ERP system are interacting mutually with each other in a cyclic process. In this paper, we introduce the core idea of the interaction approach and delineate its potentials. We discuss arising challenges in practical application and describe the current state of implementation.

1 INTRODUCTION AND MOTIVATION

In this paper we report on a technology transfer project between university and industry, combining simulation technology with commercial ERP systems.

We can on one hand define an ERP system as a software supporting enterprise tasks, functions and business processes in a holistic, cross-departmental and integrative manner (Fink et al., 2005, pp. 207 et seq.); (Jacob, 2008, pp. 1–2); (Gronau, 2010).

On the other hand we can understand simulation as a method for executing experiments, employing a model of dynamic processes in real or imaginary systems, with the aim of gaining insights that are re-transferable to the investigated original system. Discrete event simulation is characterised by erratic and punctual state transitions occurring at discrete points in time (VDI, 1993); (Page and Kreutzer, 2005).

Discrete event simulation is applied in the context of ERP systems mainly in three different ways:

- As *ERP Simulation* for the purpose of training personnel or students (Schenk and Draijer, 2004); (Léger, 2006); (Hopkins and Foster, 2011); (Cronan and Douglas, 2012); (Nisula and Pekkola, 2012);
- As *Online Simulation* for operative manufacturing support with short-term time horizon (Wu and Wysk, 1989); (Davis, 1998); (ElMaraghy et al., 1998); (Fowler and Rose, 2004); (Cardin and Castagna, 2011); (Noack, 2012);
- As classic *Offline Simulation* studies for the analysis of strategic or tactical design alternatives (Kuhn, 1998); (Košturiak and Gregor, 1999); (Bayer et al., 2003); (Page and Kreutzer, 2005, chap. 15).

Each of the three approaches mentioned above has its own focus; however none is completely exploiting the full potential of combining a general purpose, discrete event simulator with an actually operational ERP system in an enterprise:

So-called ERP Simulation hardly employs computer simulation in the narrower sense (cf. Page and Kreutzer, 2005, pp. 9 et seq.). Here, we rather

Koors A. and Page B. (2013).
Interaction of Simulation Tools with ERP Systems - Concept and Practical Implementation.
In *Proceedings of the 3rd International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, pages 229-237
DOI: 10.5220/0004595602290237

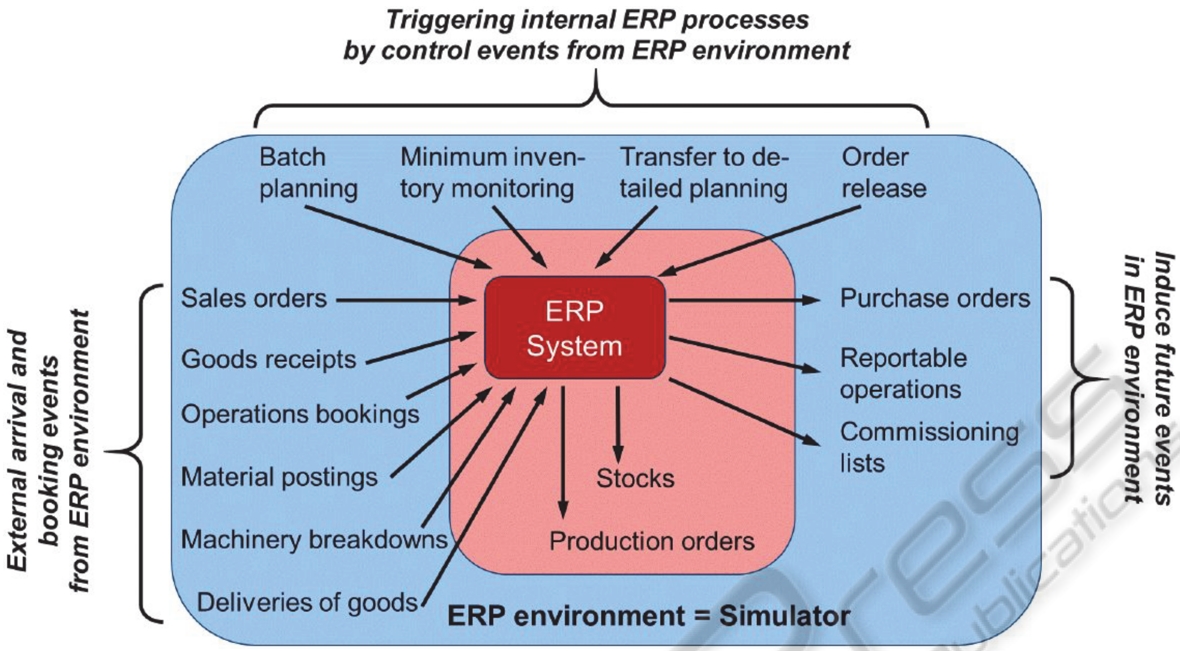

Figure 1: Structural relation between simulator and ERP system.

deal with turn-based strategy games utilising real ERP systems with pre-initialised data bases. Continual manual user intervention is desired and also required.

Online Simulation is typically reacting only on a small functional subset of ERP systems (manufacturing, production control centre, possibly medium-term scheduling) and does not necessarily automatically affect the operational ERP system. Often intervention remains reserved for the human production supervisor.

In classic Offline Simulation studies only important partial functionalities of the ERP system are modelled as operational components. For effort reasons, this is carried out mostly in an abstracted and generalised form. This restricts transferability of simulation results back into practice and limits the validity of resulting conclusions.

This paper is structured as follows: Section 2 introduces the core idea of the interaction approach. Sections 3 and 4 delineate the potentials and challenges in practical application. Section 5 describes the current state of implementation and Section 6 summarises and concludes the paper.

2 PROPOSED INTERACTION APPROACH

In the following we introduce an *Interaction Approach* between simulation and ERP systems, reaching beyond the common three approaches mentioned above for the following reasons:

- Utilisation of a full discrete event simulator;
- Coverage of all ERP functional areas;
- Use of the original ERP system algorithms;
- No data redundancy in simulator and ERP system;
- Free of manual interventions into simulation process.

The core idea of the interaction approach is as follows:

The concrete ERP system in use is copied to a new simulation instance and initialised with a data base snapshot of a given key date (e.g. today, last inventory date or start of the financial year). The simulator is simulating the complete ERP environment against the ERP system, i.e. all operationally relevant changes with impact on the ERP system, like human input and third party software interfaces (Figure 1).

This includes the arrival of new sales orders, receipt of goods purchased, booking of operations, posting of material, etc. Beyond that, the simulator is periodically triggering internal batch processes in the ERP system as it would happen in real daily operation, such as batch planning, minimum inventory monitoring, etc. Because of the modified data situation caused by the simulator and its triggering of ERP planning processes, the ERP

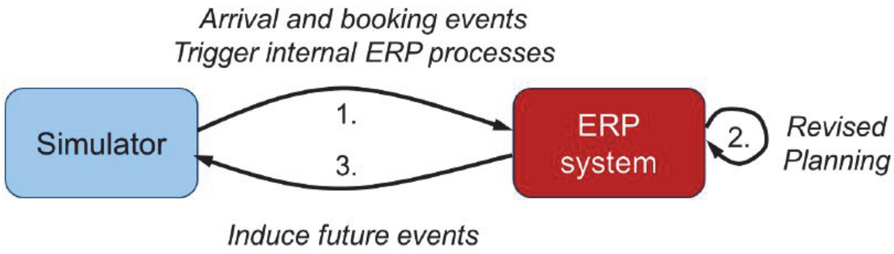

Figure 2: Flow of interaction between simulator and ERP system.

system updates and re-adjusts its plans, resulting in modified dates and quantities of manufacturing, adapted purchase order proposals, etc. Hereafter, the simulator is reacting upon the adjusted ERP system plans and is implementing them by utilising ERP system functionality, e.g. by releasing new manufacturing orders or generating new purchase orders. Furthermore, the simulator schedules arising future events like prospective operations and material bookings for a later point in simulation time. After advancing the simulation clock (and simultaneously the synchronised ERP time), the scheduled simulator events induce simulated dynamics of the ERP system environment again. All in all, the simulator and the ERP system are interacting alternately and cyclically with each other (see Figure 2).

From the ERP system point of view the simulator is representing the operational environment where it is receiving its input from. Moreover, the results of re-adjusted ERP system plans influence the simulation environment, as they would influence the real operational environment in daily operation.

Conversely, the simulation model in the simulator is describing the ERP system related operational processes in a company. Here, only the company-specific application is modelled (type, order, points in time and duration of used ERP system functions, as well as upstream extraction of company specific data for parameterisation of the simulation model). For the actual processing of ERP functionalities, however, the simulation instance of the ERP system is accessed by means of software interfaces. From the simulator point of view, the ERP system is “only” a large domain-specific software library, whose functions it is calling and whose successor states it is reading by appropriate data interfaces. By executing a model, the simulator is actually carrying out a software *automation* in the sense of complete remote control of the ERP system.

A similar approach is suggested in Herrmann (2007), however focus and implementation differ, particularly in degree of interleaving between simulator and ERP system. The work introduced here has been developed independently, in the context of industrial cooperation.

Our approach has a number of assets to offer:

- It can be widely avoided to implement complex ERP specific functionality in the simulator, e.g. scheduling logic. Instead, the available ERP system functionality is called 1:1, as on hand in a program library. Thus effort and time of model building are significantly reduced.
- Likewise, calling original ERP system functions avoids inaccurate mapping of ERP system functionality into simulation models. Thus simulation results guarantee a higher degree of validity and significance as well as close correspondence with the actual operational ERP system, allowing easier transfer back into ERP practice.
- By employing the ERP system in its entirety, questions relating to an enterprise as a whole, from sales to purchasing departments, can be analysed more adequately. Compared to common isolated simulation of manufacturing, the comprehensive view on the whole company allows for a more substantial analysis of department overlapping problems, stemming from shortcomings in inter-departmental cooperation or from the applied ERP system itself, resp. from its parameterisation.

3 APPLICATION POTENTIAL

Beyond general simulation studies requiring detailed modelling of the ERP system, our interaction approach naturally is relevant for various issues specifically dealing with ERP systems in operation, their parameterisation or their company specific utilisation. In more detail, the following application options are at hand:

a) Variation of the simulated environment: Stress factor testing as well as sensitivity analyses of the company organisation can be carried out by varying parameters of the simulation model while leaving the ERP system configuration unchanged. For instance the impacts of an extension or wider variance of material delivery times or of an increase of machine failure frequencies could be analysed. A variation of sales order profiles or order arrival frequencies allows for conclusions whether a company can serve their clients in time, even in altering market environment.

b) Variation of ERP parameters: By modification of the ERP system parameters without changing the simulation model itself, the behaviour and parameter sensitivity of an ERP system can be explored, eventually optimising ERP parameters. Obvious examples for this option are variations of lot size parameters, minimum stock levels, machine group capacities or time horizons. Beyond that, proposed modifications of the production process (changed bills of material or work plans) or the introduction of new products and technologies can be investigated.
c) Variation of ERP utilisation: By altering the business processes or called ERP functions in the simulation model, modified handling of the real ERP system can be studied, e.g. regarding impacts of proposed business process reorganisation.
d) Variation of ERP algorithms: Adaptations of internal ERP program logic can automatically be tested, driven by the simulation model and without interference with productive ERP operation. This allows assessing benefits and side effects under defined environmental conditions, on the side of the software producer (e.g. new planning, scheduling or order release algorithms) as well as of the users (e.g. company specific adaptations).

By comparison of predefined key performance indicators and analysis of relationships between variation scenarios, additional insights into company dynamics can be gained, in order to prepare or support optimising measures.

4 CHALLENGES

In order to achieve the expected benefits from the interaction approach, the following aspects have to be taken into consideration:

- Necessity of modelling: By copying the productive ERP system, simulation modelling of ERP functionalities becomes redundant. Company specific settings and adaptations in the ERP system are copied as well. By importing a current or historical snapshot of the ERP data base, no further initialisation on the ERP system side is needed. Nevertheless, the simulation model has to be adjusted to the way the concrete modelled company handles its ERP system; i.e. the business processes of ERP utilisation have to be documented and represented in the simulation model. This is an individual process for each company, requiring corresponding individual analyses and manual work in implementation, comparable to a classic simulation study.
  Our interaction approach cannot provide an adequate simulation environment ‚out of the box' that is suitable for each company without modification.
- Experiment duration: A simulation experiment covering for instance one financial year of a company is requiring in principal at least the same computing time as the sum of all called ERP system functions would consume in daily operation during the simulated time, assuming simulation is performed on the same hardware. Additional overhead for processing the simulation model itself has to be added, and a certain experimentation factor has to be taken into account; i.e. ERP functions run partially concurrently on network clients in real company operation, whereas in simulation experiments these functions may be called sequentially only on one client or server, unless the simulation model itself is implemented in a true concurrent manner.
  Very significant factors are time-consuming ERP batch runs like daily batch planning. With a realistic batch planning time requirement of 1 hour per day, the simulation of one financial year (260 working days) will result in a time consumption of at least 260 hours per experiment, corresponding to 11 calendar days per experiment. In addition, experiment replications with varying starting values of random number generators are indispensable for statistical reasons. For efficiency considerations, this suggests parallel execution of simulation experiments (Illner, 2013) and therefore parallel operation of multiple ERP system copies for simulation purposes.

The general value of each simulation study is highly dependent on data quality. This is also true for our interaction approach. Thus, the following aspects have to be taken into account:

- The underlying ERP system data may be incomplete, because not all ERP relevant business processes are handled via the ERP system. The degree of ERP system implementation in daily operation may depend on how long the ERP system is in practical use, likewise on the expected utility of using or not using certain ERP system functions. For instance, purchase orders of auxiliary materials and consumables could still be handled outside the ERP system, or messages on machine malfunction could be waived. Accordingly, the statistical data basis for simulation could be incomplete.

- A related problem arises if the ERP system is not utilised continuously or not in a consistent manner, i.e. users may use the ERP system only partially or carry out different ERP system operations for the same task. If business processes with identical content are treated in different ways in the ERP system, semantically equivalent data may be fragmented in the data base. These aspects can lead to seemingly missing or inconsistent data and wrong conclusions about utilisation frequency. For instance, inconsistent or mingled use of stock correction bookings and inventory postings will communicate a wrong image of the real inventory process. In such cases the statistical data basis for simulation would be incomplete or distorted and therefore invalid.
- Possible shifts of model structure in timeframes used for statistical data extraction may be an issue in every simulation study; however here they have to be examined with special care, because companies indeed are subject to permanent change and continuously aiming at optimisation. Business processes and thus ERP system utilisation have a high potential for alteration during the time interval used for statistical data extraction: For instance introduction of new market segments, opening or closing down of production sites, commissioning of new production technologies or other internal process reorganisation may lead to certain additional, different or missing data in the ERP system data base. Such structural discontinuities have to be accounted for, as the data generating process has changed during the analysis period and we cannot act on the assumption of identically distributed data anymore. Thus, ignoring existent development of a company will lead to statistical problems, e.g. due to mistaken multimodal distributions.

Beyond the statistical data basis issue, we have to model the individual company utilisation pattern of the ERP system. In this context the following aspects have to be considered:

- When modelling ERP utilisation for simulation, it is essential to systematically investigate those ERP relevant functions which are deliberately handled outside the ERP system or which are in principle not covered by the given ERP system functionality. These business processes have to be reproduced in the simulation model by complementing the model with a formal representation of the original human handling process. This might be very labour-intensive, especially if the reason for non-utilisation of the ERP system arose from the complexity of a task whereas the ERP system producer was unable to provide an adequate solution. Examples could be scheduling and order release priorities or short notice re-scheduling of orders due to external events or unexpected instructions from higher management. If necessary, additional automation interfaces to indispensable third-party IT-systems have to be implemented in case expertise has been outsourced, e.g. to APS or MES.
- In order to promote an initial trust into the constructed simulation model it makes sense to initialise the simulation copy of the ERP system with a historical data base snapshot (e.g. beginning of the previous calendar year) and to afterwards run the simulation to a second past point in time (e.g. beginning of the current calendar year). An impression on the model quality and its dynamics can be gained by comparison of the final simulation state with an independent ERP data base snapshot of the second point in time, where the simulation ended. However, dynamic deviation of real ERP system utilisation, as contrasted with static use patterns modelled for simulation, might turn out problematic, but may be more likely here than in many other domains. In this instance, model validation based on historical data might fail and following simulation results can be questioned. For validation purposes, any rapid or subtle changes in ERP utilisation patterns over the course of time have to be detected and replicated in the model, which may require considerable effort. At worst, comprehensive modelling of dynamics might become impossible, if altering ERP system utilisation is not documented, not specifiable or simply unaware in the modelled company.

5 IMPLEMENTATION

Our interaction approach described above has been implemented in graduate thesis and master projects at the University of Hamburg (Kühnlenz, 2011); (Schäfer, 2011); (Reichelt, 2012), using the open source simulation framework DESMO-J developed at the University of Hamburg (Page, 2013) as well as the commercial ERP system ERP COM of Infor GmbH (Infor, 2013). For practical substantiation and review of the concepts introduced, we established a cooperation with two German medium-sized manufacturing companies, which have been using ERP COM operationally for many years.

The choice of Infor's ERP COM system was motivated by easy academic accessibility, very suitable student training material, a broad customer base for potential cooperation and already existing program know-how. In contrast and corresponding to DESMO-J's free availability, interaction with an open source ERP system could have been an appropriate alternative, establishing an affordable combination of tools for innovative small and medium size enterprises. However, currently open source ERP systems have a rather low market penetration in Germany and often lack reliable support (Borgmann, 2010). In order not to complicate the project more than necessary, it was decided to interact with an established ERP system.

First, we turned our attention to development of a technical interface to the ERP system data base. The interface permits to read actual arrival and booking events of the ERP system within a selected concrete historical interval of time. Using the empirical data collected, parameters for diverse random distribution type candidates are estimated by an R script. Subsequently, the final distribution types are identified, making use of the Kolmogorov-Smirnov and Anderson-Darling goodness-of-fit tests.

In this way, 8 distributions with their parameters are estimated from concrete ERP system data, e.g. for sales orders, operation durations and purchase order delivery times (Schäfer, 2011). A further tool allows visualisation of these distributions directly from the utilised ERP system. Here, the derived stochastic distributions are presented superimposed by the actual ERP system data. This validation step serves as visual check as well as for transparency and confidence building for model users, concerning the quality of simulation model input. The tool provides further added value by facilitating adjustment of independent ERP system parameters w.r.t. actually observed data in ERP system application, e.g. comparison of operation duration master data vs. practically observed operation duration (Reichelt, 2012).

As a second step, a reference simulation model has been implemented in DESMO-J, reproducing the standard business processes of using ERP COM in practice, as advised by its producer Infor (Infor, 2012). Here, all standard business activities from sales to planning, manufacturing, warehousing and purchasing have been included. Altogether 15 functional interaction points have been identified, where the simulator is calling ERP system functionalities, e.g. creation of sales orders, triggering of planning, release of manufacturing orders, generation of purchase orders, booking of goods receipt or manufacturing operations, etc. The simulation model is accessing the ERP system via abstracted technical interfaces in order to create business objects like sales orders or call ERP functions on existing business objects, just in the same way a human operator would do in the modelled company. In this context the simulation model is controlled by the 8 extracted statistical distributions described above. Additional 9 model control parameters can be specified by XML-files or are directly read out of the ERP system's data base.

Beyond that, we provide a replay mode, accessing original object sequences from the analysed ERP system data base instead of artificial random numbers. Thus, historical sales orders along with their empirically observed inter-arrival times can be re-used in a 1:1 manner in simulation experiments, as an alternative to stochastic distributions. This is in particular useful for both simulation model validation by means of output comparisons and later model parameter calibration.

The comprehensive reference simulation model has been extensively documented in textual form as well as with BPMN 2.0 diagrams (Kühnlenz, 2011).

In parallel 15 entry points for functional interaction have been created in the ERP system, allowing for automation of the ERP system by the simulator. A newly constructed control process in the ERP system is receiving simulator Remote Procedure Calls, including parameters, via a TCP/IP socket interface. According to the received message content, the control process branches into ERP functionalities by calling and processing standard business logic of the ERP system. Meanwhile, the simulator waits until the called ERP functionality is processed completely. Hereupon the ERP control process communicates possible return values or status messages back to the simulator, again via TCP/IP sockets (Schäfer, 2011). In addition, the simulator can access the ERP system data base via an abstracted interface in read-only mode, in order to access new or modified business objects of the ERP system effected by preceding ERP remote calls.

The simulator permanently is synchronizing the local time of the ERP system by setting the computer's resp. virtual machine's wall clock time to its own internal simulation clock. After execution of ERP functionality by remote procedure calls, the simulation clock is not adjusted to the current wall clock time, because events already scheduled on the simulator's event list could be skipped, especially in cases of long ERP batch runs. By this means, execution of ERP functionality does not consume simulation model time and appears atomic. This deviates from reality, because race conditions and transaction collisions will not be observed in simulation experiments. A refined implementation could delegate every ERP function call to an own processor thread and synchronise the simulation clock with wall clock time as long as ERP function calls are executed, resulting in true realistic concurrency with respect to the ERP system. However, it is questionable if this effort would be justified in the presence of a generally stochastic simulation. In summary, the ERP system is jumping from time point to time point along with the simulator in a synchronous manner, reaching processing speed as fast as possible by employing the next-event simulation approach (Kühnlenz, 2011).

The simulation model has been implemented in an event-oriented fashion, mainly for two practical reasons: First, we aimed at keeping the run-time overhead attributable to model execution down, as experiment duration is a non neglectable issue. Second, we envisage enhancing experiment handling by deploying an extension of DESMO-J that permits saving, loading and resuming event-oriented experiments at any simulation run time instant (Janz, 2010). When taking and restoring ERP data base snapshots simultaneously, interesting intermediate states of ERP simulation experiments can be preserved. By duplicating saved intermediate states and varying their parameters upon restart, alternative scenarios can be studied. In these cases, considerable time for re-obtaining intermediate states of interest can be saved, since experiment execution ab initio is avoidable. Due to restrictions of the Java standard VM, this technique is not feasible for the process-oriented paradigm, which is implemented in DESMO-J on basis of Java threads.

Modelling a company in an event-oriented style, from a bird's-eye view, seems natural in many cases, as often self-contained impersonal events like order arrival, operation booking or triggering of batch processes occur. However, it can be argued that human utilisation patterns of ERP systems may alternatively suggest process-oriented modelling, as far as company business processes are concerned and modelling is performed on a departmental level. Even one step further, agent-based simulation, as provided e.g. by DESMO-J's extension FAMOS (Knaak, 2002) could be considered, if individual roles in the company organisation are in focus and fine-grained data and detailed description of role behaviour is available at an acceptable effort. However, in this first approach we remain with event-oriented modelling, for the reasons given above.

At the end of a simulation experiment, DESMO-J can create a trace file during the simulation run, recording the course of simulated business processes. This is an important basis for model validation. Additionally, first statistical results are accessible in the standard simulation report, e.g. for queues used in the simulation model. For comparison and evaluation of simulation results, further 9 company key performance indicators (KPI) have been defined and partially implemented, e.g. adherence to delivery dates and average delay of purchase orders. Here, data from simulation runs are combined with the ERP system's data base. The aggregated KPI are output in textual form in a report file and are also visually presented in a graphical tool (Reichelt, 2012).

Apart from minor open work concerning material posting, the cyclic interaction of simulator and ERP system is completely implemented.

6 SUMMARY AND CONCLUSIONS

In contrast to conventional approaches of linking ERP systems with discrete event simulators, our interaction approach introduced in this paper aims at coupling a full-fledged discrete event simulator and a copy of an ERP system in full productive operation, fully integrating comprehensive ERP functionality as used in real, daily practice and without human manual intervention. Here, the simulator is representing the complete operative environment of the ERP system, substituting its daily business input. For this, the company-specific utilisation of the ERP system (not the ERP functionality itself) is modelled in the simulator. In order to execute its model and process concrete ERP functionality, the simulator accesses the ERP system via software interfaces, using it like a large subject-specific software library. Thus, the simulator is effectively carrying out a complete remote control of the ERP system, in the sense of software automation.

The simulator causes arrival and booking events in the ERP system, is triggering internal ERP processes on a regular basis and is processing results like revised planning and adjusted orders by scheduling future events, according to stochastic distributions extracted from the ERP system beforehand. Altogether simulator and ERP system are interacting mutually with each other in a cyclic process.

By using this concept, only very few ERP functionality has to be represented in the simulation model, allowing for a faster and more precise model construction with less effort. Simulation results are more trustworthy and easier to re-transfer into ERP practice.

Our approach represents a comprehensive and integrated simulation of business processes and is in particular predestinated for issues where concrete ERP systems in real company operations play a significant role. Application potential is on one hand arising from variation of the simulated environment, e.g. for company sensitivity analyses or load tests of manufacturing. On the other hand, variation of ERP parameters allows exploring the behaviour and sensitivity of the ERP system, in order to eventually optimise ERP parameters. Furthermore, the simulation of altered utilisation of the ERP system can help analyse alternative business processes, whereas variations of ERP algorithms allow for examination of modified program logic in a safe, controlled and dynamically reproducible experimental environment

In order to assess the utility of the interaction approach, the length of simulation experiments has to be regarded. Comparable to a classic simulation study, individual modelling of ERP utilisation is necessary, even if a copy of the real ERP system with extensive ERP functionality is at hand. It has to be stressed that trustworthy simulation results can only be expected if the ERP data basis and the modelled ERP utilisation processes are complete, consistent, valid and representative. Ensuring this may require considerable effort.

The interaction approach has been implemented using the open source simulation framework DESMO-J developed at the University of Hamburg in conjunction with the commercial ERP system ERP COM of Infor GmbH. Implementation was carried out in cooperation with two medium-sized manufacturing companies, in the context of graduate thesis and master student projects.

At the current status of our project, company specific ERP model parameters are extracted automatically from the ERP data base and control a comprehensive, well documented event-oriented simulation model under DESMO-J. The simulation model is automating the ERP system via newly generated technical interfaces. The results of triggered ERP processes affect the simulation model which in turn is reacting with new external impulses. In addition, tools for measurement and visualisation of company KPI have been implemented.

The interaction cycle as presented in this paper is basically operational. As a next step, some fine tuning and software additions are required as prerequisite for real-world simulation experiments in the ERP environments of the two cooperation companies.

Once the technical basis is established, it carefully has to be dealt with methodical challenges regarding data quality, model accuracy and run time. Further validation is needed before concrete ERP related operative and tactical investigations like ERP parameter tuning, load tests and analysis of alternative ERP utilisation can be addressed.

ACKNOWLEDGEMENTS

The authors would like to thank the company Infor GmbH, the Infor User Forum, the two industrial cooperation partners, the PML Section of Department 4 at the FH Düsseldorf as well as the Computing Center of Hamburg University for their support, software, ideas and contacts. We also want to express our recognition of the strong working contributions of our graduate students Claas-Michael Kühnlenz and Fabian Schäfer.

We appreciate the comments of two anonymous referees of this conference, whose advice has been incorporated into the body text.

REFERENCES

Bayer, J., Collisi, T. and Wenzel, S. (2003), *Simulation in der Automobilproduktion*, Springer, Berlin.

Borgmann, S. (2010), “Open-Source-ERP-Systeme im praktischen Einsatz in kleinen und mittleren Unternehmen (KMU)”, Master's Thesis, Carl von Ossietzky Universität Oldenburg, Oldenburg, 2010.

Cardin, O. and Castagna, P. (2011), "Proactive production activity control by online simulation", *International Journal of Simulation and Process Modelling*, Vol. 6 No. 3, p. 177.
Cronan, P. and Douglas, D. (2012), "A Student ERP Simulation Game: A Longitudinal Study", *Journal of Computer Information Systems*, Vol. 53 No. 1, pp. 3–13.
Davis, W.J. (1998), "On-Line Simulation: Need and Evolving Research Requirements", in Banks, J. (Ed.), *Handbook of Simulation*, John Wiley & Sons, Inc, Hoboken, NJ, USA, pp. 465–516.
ElMaraghy, H.A., Abdallah, I.B. and ElMaraghy, W.H. (1998), "On-Line Simulation and Control in Manufacturing Systems", *CIRP Annals - Manufacturing Technology*, Vol. 47 No. 1, pp. 401–404.
Fink, A., Schneidereit, G. and Voss, S. (2005), *Grundlagen der Wirtschaftsinformatik: Mit 16 Tabellen, Physica-Lehrbuch,* 2nd ed., Physica-Verl., Heidelberg.
Fowler, J.W. and Rose, O. (2004), "Grand Challenges in Modeling and Simulation of Complex Manufacturing Systems", *SIMULATION*, Vol. 80 No. 9, pp. 469–476.
Gronau, N. (2010), *Enterprise Resource Planning: Architektur, Funktionen und Management von ERP-Systemen,* 2nd ed., Oldenbourg, München.
Herrmann, F. (2007), "SIM-R/3: Softwaresystem zur Simulation der Regelung produktionslogistischer Prozesse durch das R/3-System der SAP AG", *WIRTSCHAFTSINFORMATIK*, Vol. 49 No. 2, pp. 127–133.
Hopkins, J.L. and Foster, S. (2011), "Supply Chain ERP Simulation: A unique learning experience", paper presented at AMCIS, 4-8 August, Detroit, Michigan, USA.
Illner, M. (2013), "Konzeption und Implementation einer Batch-Verarbeitung von Experimentreihen in DESMO-J", Bachelor's thesis, University of Hamburg, Hamburg, 2013.
Infor (Deutschland) GmbH (2012), *Training material on ERP COM 7.1*.
Infor (Deutschland) GmbH (2013), *ERP COM 7.1: ERP-System. http://www.infor.de/produkt-spotlight/erp/erp-com*.
Jacob, O. (2008), *ERP Value: Signifikante Vorteile mit ERP-Systemen, Xpert.press*, Springer, Berlin, Heidelberg.
Janz, T. (2010), "Prototypische Implementierung einer Verwaltung von Momentaufnahmen im Simulationsframework Desmo-J", Bachelor's Thesis, University of Hamburg, Hamburg, 2010.
Knaak, N. (2002), "Konzepte der agentenbasierten Simulation und ihre Umsetzung im Rahmen des Simulationsframeworks DESMO-J", Master's Thesis, University of Hamburg, Hamburg, 2002.
Košturiak, J. and Gregor, M. (1999), "Simulation in production system life cycle", *Computers in Industry*, Vol. 38 No. 2, pp. 159–172.
Kuhn, A. (1998), *Simulation in Produktion und Logistik: Fallbeispielsammlung*, Springer, Berlin.
Kühnlenz, C.-M. (2011), "Interaktion von Simulationswerkzeugen mit ERP-Systemen - Konzeption und Realisierung von Interaktionsworkflows am Beispiel von DESMO-J und Infor ERP COM", Master's thesis, University of Hamburg, Hamburg, 2011.
Léger, P.-M. (2006), "Using a Simulation Game Approach to Teach Enterprise Resource Planning Concepts", *Journal of Information Systems Education*, Vol. 17 No. 4, pp. 441–448.
Nisula, K. and Pekkola, S. (2012), "ERP-based simulation as a learning environment for SME business", *The International Journal of Management Education*, Vol. 10 No. 1, pp. 39–49.
Noack, D. (2012), "Online Simulation in Semiconductor Manufacturing", PhD thesis, Fakultät für Informatik, Universität der Bundeswehr München, Munich, 2012.
Page, B. (2013), *DESMO-J: Simulation framework. http://www.desmo-j.de*, Universität Hamburg, Fachbereich Informatik, Modellbildung und Simulation.
Page, B. and Kreutzer, W. (2005), *The Java simulation handbook: Simulating discrete event systems with UML and Java*, Shaker, Aachen.
Reichelt, D.G. (2012), "Interaktion von Simulationswerkzeugen mit ERP-Systemen - Konzept und Umsetzung zur Erweiterung und Vervollständigung der bestehenden ERP-Simulation", Report on Master's project, University of Hamburg, Hamburg, 2012.
Schäfer, F. (2011), "Interaktion von Simulationswerkzeugen mit ERP-Systemen - Konzeption und Realisierung von Datenanalysen sowie technischen Schnittstellen am Beispiel von DESMO-J und Infor ERP COM", Master's thesis, University of Hamburg, Hamburg, 2011.
Schenk, D.J. and Draijer, C. (2004), "Best Practices of Business Simulation with SAP R/3", *Journal of Information Systems Education*, Vol. 15 No. 3, pp. 261–265.
VDI Verein Deutscher Ingenieure and Gesellschaft Fördertechnik, Materialfluss und Logistik (1993), *Simulation von Logistik-, Materialfluß- und Produktionssystemen: Simulation of systems in materials handling, logistics and production, VDI-Richtlinien No.* 3633, VDI-Verlag, Düsseldorf.
Wu, S.-Y.D. and Wysk, R.A. (1989), "An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing", *International Journal of Production Research*, Vol. 27 No. 9, pp. 1603–1623.