Loose Model Coupling by Iteratively Driven Simulations for a Model Pipeline

Jochen Wittmann[1], Kai Himstedt[2]
[1]HTW Berlin, FB2, Umweltinformatik, Wilhelminenhofstr. 75A, 12459 Berlin
wittmann@htw-berlin.de
[2]Universität Hamburg, Department Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg
himstedt@informatik.uni-hamburg.de

Abstract

In the context of the cooperation project “Efficient Airport 2030” the classical task arises to realize a model coupling between some complex and methodological different models with the final intention to be able to answer the questions originating in the over-all method of a scenario analysis. Each phase from the home of the passengers to the take-off of the aircraft is modeled by a separate sub-model, each with its own modeling paradigm. In regard to the relatively loose interconnections between these models and the forward-oriented pipelining character of the data flow there is no need for a close coupling especially with respect to the limited demands of the intended model interpretation given by fixed scenarios. A common time handling for the submodels can neglected, therefore. Nevertheless, a basic feedback mechanism in the model pipeline is necessary, which is implemented by the approach of iteratively driven simulation. This approach results in a significant increase of simulation runtime but it has the advantage to leave the coupled submodels completely unchanged as independent black-boxes that need not be extended for additional communication.

1 Introduction

In the context of the cooperation project “Efficient Airport 2030” the complete workflow at an airport without those parts being under the control of the air security shall be modeled and optimized. The project is founded by the German “Bundesministerium für Bildung und Forschung (BMBF)” with respect to the growing air traffic. The predicted increase in air traffic causes innovative solutions in infrastructure on the one hand but on the other hand also in innovative concepts for an economical, ecological, comfortable, safe and flexible air transportation. The time horizons for the planning are the years 2015 and 2030. The project group consists of universities, research centers and industry working in the areas of air-traffic management, logistics, aircraft construction, system development and system simulation. Partners are the University of Hamburg, the Technical University of Hamburg Harburg, the University for Applied Sciences Hamburg, the German Aerospace Center (DLR), the airport Hamburg and the industrial partners Airbus, Siemens, AlsterAero, mb+Partner and German Airport Consulting (see in more detail in the project web pages under “Spitzencluster Hamburg 2009” [6]).

The final aim for the project and thus for the model coupling concept explained in this paper is to describe the process chain beginning at the home of the passenger, the way to the airport, the way of passenger and luggage within the airport area, the ground handling of the airplanes, the boarding, the loading, and finally the take off. The models will have to represent technical and organizational developments predicted by the scenario description and shall evaluate the implications of these developments on the throughput, the comfort, and the impact on other relevant ratios for the air transportation in general. Fig. 1 depicts the airport and the main processes being linked in the scenarios.

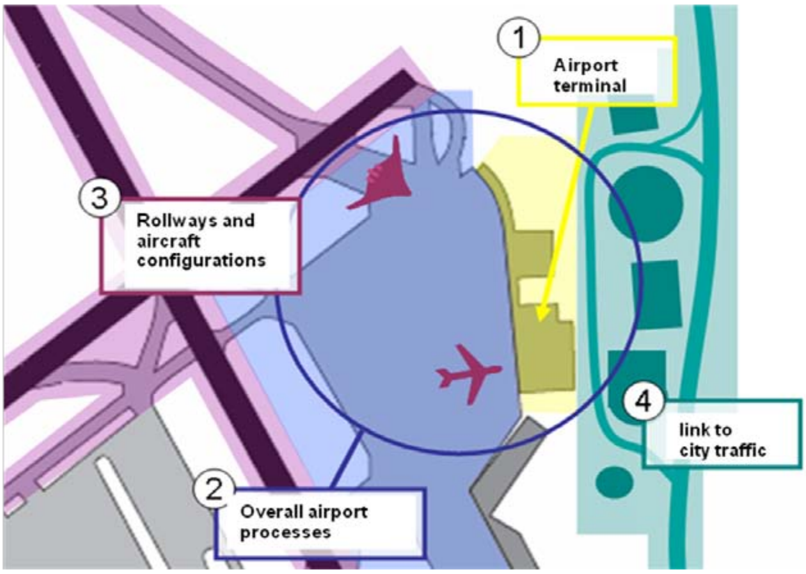

Fig 1. main groups of processes at the airport

The organization of the airport as a set of coupled subsystems enables a detailed analysis of the separate process steps in their current representation as well as in different future modifications and ameliorations to a) increase the efficiency of the transportation to the airport, b) to be able to test different means of passenger guidance within the terminals (e.g. by digital boarding cards), and c) to optimize the ground handling by new organizational but also by new aircraft configurations.

The following submodels are developed:

- M1: The passengers' ways from home to the entrance of the airport is realized as an individual-based traffic generation model.

- M2: from entrance of the terminal to boarding with special respect on check-in process, passengers path-finding and the luggage flow

- M3: model for landings, rollways and starts.

The main reason for this organization of the project lies in the fact that the different work packages can be implemented by different project partners each specialized on certain aspects in modeling and holding detailed system knowledge for the relevant process step. The experiences of the partners are substantiated in already existing (quite complex) models, partially executable on special hardware platforms only. These preconditions imply two demands: First, the demand for the autarky of the submodels in the architecture and second, the demand for real distributed execution of the model runs (because of the special hardware needed).

In Wittmann et al. 2009 [10] the basic idea for an integration of the submodels under the conditions sketched out here is described in some more detail. For the implementation of the pipeline concept a web-based structure has been chosen on the base of Web Services [11] and the XML context as a de facto standard for exchanging large amounts of data in heterogeneous environments. This pipelining concept is based on a feedback-free forward oriented dataflow in the model pipeline. Each model gets its data from its predecessor in the line, executes its calculations and afterwards its output serves as input for the following element in the chain. The data exchanged may be characterized by time stamps but there are neither any feedback connections nor any communications during the calculation phase of the segments in the pipeline. For the stream of passengers to be modeled, this seemed to be sufficient. But during the development of the scenario descriptions the demand to model very special feedback situations came up.

This paper focuses on the extension of the pipelining concept to realize those feedback situations under the restriction of holding the black-box character for the submodels in the pipeline. Before the approach shall be introduced a short view on the scenario technique in general in the context of modeling studies shall be given.

2 Scenario Analysis

The scenario analysis is a method developed by business sciences with the aim to derive prognosis for system behavior under given well defined alternatives in system configuration and parameterization. In a first phase the objectives and the problem has to be analyzed to get a list of relevant parameters and influencing factors. The next step evaluates the interrelations between these factors and develops possible scenarios for future developments specified by the factors selected. The scenario analysis will not be able to predict the future in general, but under the chosen premises the probability for the predicted behavior is reasonable high (for more details see e.g. Gausemeier et al. 2009 [2]). For the project "Efficient Airport 2030" the following global scenarios are used that are developed by the Advisory Council for Aeronautics Research in Europe (ACARE) and that are described in Phleps [3]:

A) Constrained Air Traffic Growth: 3% annual economic growth, political stability, no increase in security matters, increasing energy costs, increasing environmental consciousness, increasing demands for quality in transportation.

B) Segmented Business Models: 4% annual economic growth, increased political stability, increased energy demand, exploitation of new energy sources, moderate environmental consciousness, moderate security demands

C) Block Building: 1% annual economic growth, political tensions between EU and USA and within Asia, strongly higher energy costs, low environmental consciousness, high security demands

On the base of these scenarios Phleps (also in [3]) derives a set of 20 influencing factors and their corresponding initial values by which the parameters of the model pipeline for the simulations can be set.

3 Model Coupling

Between the three models and the resulting five submodels in the model pipeline, the following system interfaces can be elaborated. Fig 2 shows the components and the connecting flows thus demonstrating the pipeline character of the coupled model architecture.

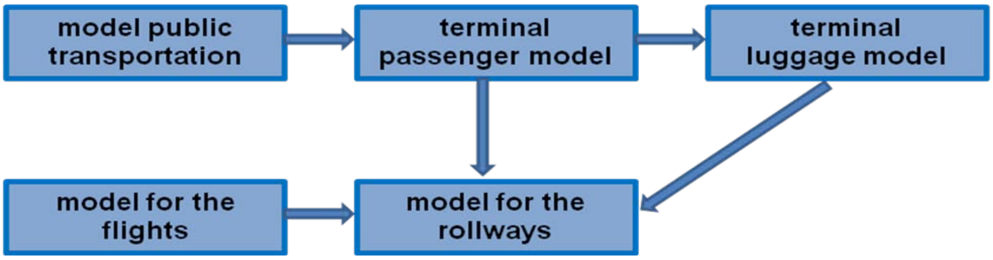

Fig.2: the model pipeline with the submodels of the project groups

The evaluation of the scenarios is realized by a calculation of the submodels along the pipeline. As already explained, the black-box-character of the submodels should be guaranteed because of the organization of the projects work packages and their distribution on the partners. Therefore, the realization of the model architecture by the means of commercial software solutions for distributed and parallel simulations does not meet the demands of the project context. Strassburger et al. apply for example the High-Level-Architecture (HLA) for applications with main focus on reusability of the submodels. In this study (see Raab et al. in [4]) however, all development was made on the base of a monolithically usable version of a single source code, a precondition that conflicts with the demand for black-box-philosophy in the airports model pipeline. Main point for using the HLA mechanisms is always a common handling of the time to provide well defined access to common data and correct inter-component communication (see Straßburger in [7]). For a detailed discussion of different approaches to couple simulation models in an distributed environment see Wittmann et al. in [10] together with Bach et al. in [1] and Rohwer in [5].

The pipeline character of the model chain handled in the project context leads to the solution of a pipelining concept for the model execution, too, that is introduced in [10]. This concept fulfills all the demands of the scenario analysis and it has the special advantage that even during model execution the autarky and independency of each submodel of the pipeline is assured. Fig. 3 shows the pipeline-coupling in contrast to the classical coupling.

For the example with the models M1, M2, and M3 it is shown that the execution time intervals overlap each other as well as the access to at least one common data, the current value of the simulation time t, has to be managed. This task is solved by a common runtime system organizing the synchronization points between the submodels together with the access to the global data. Using the pipeline concept, the execution intervals of the submodels do not necessarily overlap, the models are calculated one after the other. On the data level the relation between the submodels M1, M2, and M3 is a simple "before"-relation. This implies that the data of model M(i) has to be completely calculated before the activation of model M(i+1) can start. Naturally, such a strong chaining is in contrast to any feedback between the models in the chain, but this does not mean that within each of the models any feedback is prohibited as well. An internal feedback is possible, but it has to use only those data that already are provided by the corresponding predecessor.

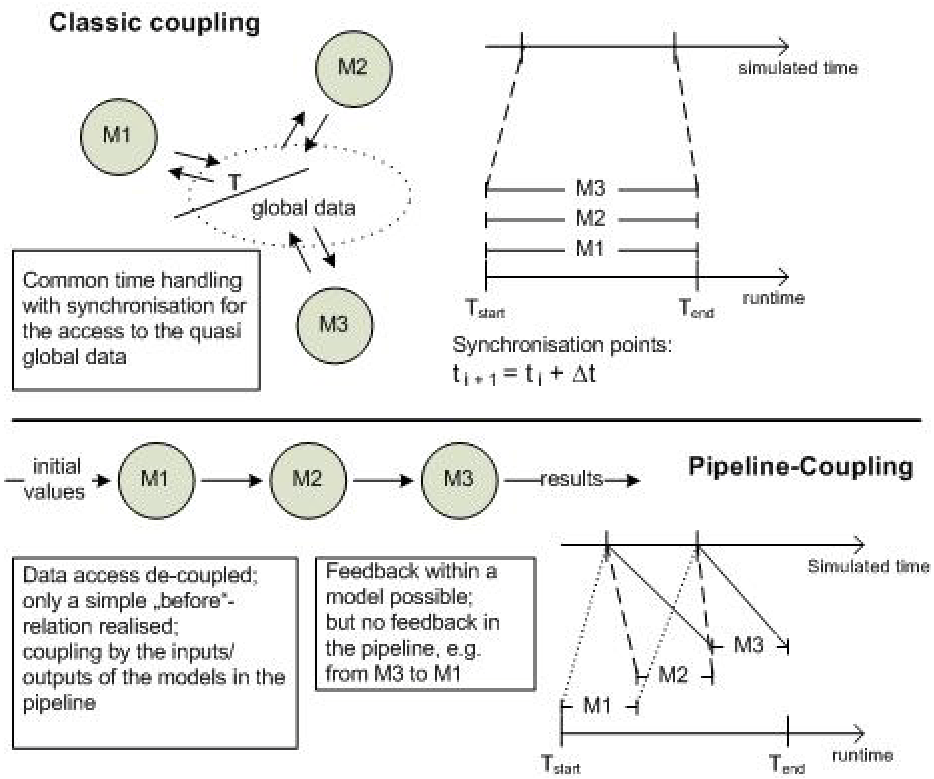

Fig. 3: the pipeline-coupling in contrast to the classic coupling

Fig 3 shows as a typical implementation of the model pipeline the coupling of model M2 (check-in up to ground handling of the aircraft) with model M1, that provides data concerning the "load" for the airport, that is the number of passengers arriving at the terminal, and the model M3 with the specification of the air-side and the take-off.

For the passenger model the stream of incoming passengers is distributed on the different stations with their corresponding queues within the terminal (check in, luggage, pass control …). This passenger stream is divided in time segments of one hour resulting in a table with the number of passengers arriving each hour of the day. The time resolution is not a restriction of the pipeline but is the typical resolution of the modeling paradigm used and the output data generated by the model M1. Passengers arrive during the day with the typical maxima in the morning hours and in the evening. All statistical data for validation is collected in one hour time steps, implying that any finer scaled statistic would be inadequate.

As long as there is no need to model feedback-loops as it suggests the forward-oriented structure of the passenger-flow depicted in the figure 2 a simple model pipeline coupling the submodels would be sufficient and could be implemented in a distributed software environment as the project context demands. But during experimentations with this pipeline, scenarios can be imagined that exceed the functionality of this architecture and make a feedback within the data-exchange streams between the submodels necessary. For example there might happen a special event in the time interval between 15:00 and 19:00 o'clock (e.g. some technical problems that is caused by the rollway processes) causing a delay for the take-offs in this period of about 2 hours. This event is modeled in the submodel M3. Additionally one has to assume that the airlines and/or the airport are able to inform their passengers about this delay (e.g. just by the modern devices to be under consideration in the scenarios) causing some reactions concerning their way to the airport. This is the feedback of submodel M3 to submodel M1; the pipeline as explained so far would not be capable to handle such feedback data flows within a single simulation run.

In a first approach, the problem might be solved by introducing additional functionality like "state-saving" and "rollback" within the submodels and an organization of the feedback situation as a re-simulation starting with the point in time the data arrives in the preceding submodel. However, this solution presumes firstly that feedback relevant situations can be identified within the pipeline and secondly that all the submodels provide the functionality to save their complete internal state at any given point in time and to administer these states within a simulation. As found Straßburger et al. in [8] commercial simulation systems do not support such demands sufficiently. For the project context, the administrational efforts within the submodels and the necessary communication between them concerning the resetting in case of feedback situations would imply an infringement of the postulated autarky and the black-box character of the submodels. Therefore, another approach was chosen to solve the feedback.

4 Iteratively Driven Simulation

The approach to be introduced here utilizes the raw scale for data exchange of the pipeline for the airport scenarios. The time interval for collecting data is 1 hour, because all statistical data measured in the real system and needed for verification and validation is only available in this scale. Based on this premise the feedback interval is at least 1 hour long and simulating the course of one day in the scenarios, the overall simulation run consists of 24 potential interacting and/or feedback points maximum.

Simulation time T, with Δt=1h

	0 .. 1	**1 .. 2**	**2 .. 3**	**3 .. 4**	**4 .. 5**	**5 .. 6**	**6 ..7**	**7 .. 8**	**-1 ..**
Run 1	Initial values→ M1→M2→M3→ E[0..1]								
Run 2	Initial values, E[0..1]→ M1→M2→M3→E[0..2]								
Run 3	Initial values, E[0..2]→ M1→M2→M3→E[0..3]								
Run 4	Initial values, E[0..3]→M1→M2→M3→E[0..4]								
Run 5	Initial values, E[0..4]→M1→M2→M3→E[0..5]								
Run 6	Initial values, E[0..5]→M1→M2→M3→E[0..6]								
Run 7	Initial values, E[0..6]→M1→M2→M3→E[0..7]								
Run 8	Initial values, E[0..7]→M1→M2→M3→E[0..8]								
Run	Initial values, E[0.. -1]→M1→M2→M3→E[0..]								

Given simulation time T=24h, effort for simulation one Δt=1h given by 1*SE*
Global effort for iteratively driven simulation (24 runs) makes 300*SE*
(Remark: in general given by n*(n+1)/2, with n=24 makes 24*25/2=300)

With these restrictions the basic idea is to interpret the start time for a simulation as the only restart point even in case of feedbacks. Therefore, if a feedback situation happens, the complete pipeline has to be recalculated beginning with the time, the feedback data reach the first relevant member in the pipeline. This implies an execution scheme of the simulation for all the members in the pipeline as figured in figure 2. This means in detail for the example consisting of M1, M2 and M3: In a first run all submodels are initialized with the scenario settings and simulated over the time interval [0;1] from =0:00 to 01:00 o'clock. The simulation result is defined as E[0..1]. For the following time interval this result can be offered as an additional initialization dataset.

These data are especially accessible by model M1 giving M1 the opportunity to react in the sense of a feedback on the results simulated in [0..1]. In addition, M1 can offer the results to M2 for the second run thus giving M2 information coming from E[0..1] and thus indirectly realizing a feedback even from M3 to M2 if necessary.

Naturally, the length of a run increases successively from run 3 up to 8 for 1 hour. In general, the results E[0..(i-1)] can accessed in run i even if these results are produced by submodels that have a higher rank in the pipeline. It is obvious that the access to feedback data is only possible for a time interval in the past, meaning for the example with at least 1 hour delay. This means in the example that M1 is not allowed to access data generated by M3 for the interval 15:00 to 16:00 o'clock to calculate the time interval 10:00 to 11:00. By this restriction events lying in the future are not accessible otherwise causing inconsistencies in the runs.

If we continue the example with a delay between 15:00 and 19:00 o'clock by events happening in M3, M3 now could signalize in run 16 for the time interval 15:00 to 16:00 the delay in the results E[0..16]. Beginning with run 17 and higher M1 is able to integrate these data in its own simulation for calculation the time intervals 16:00 to 17:00 and higher. Starting at 16:00, M1 will reduce the number of arriving passengers at the airport in accordance with the additional given feedback information.

By the repeated restarts of the simulation that represent the only possible reset points, the black-box-character of the submodels in the pipeline is assured. This advantage in architecture and algorithmic is paid by a considerable increase in calculation time and some effort for data transport between the elements of the pipeline. The amount of calculation time can be expressed by the formula as a summation over the time intervals handled ($n*(n+1)/2$). In the example, a single run over the 24 hour period with 1 hour scaling is calculated by the iteratively driven simulation with 300 simulation-time-units. For comparison, the reference run would need only 24 simulation-time units. This makes a factor of 300/24=12,5 for calculation time. In general, the calculation time needed is expressed by (T/deltaT + 1)/2, with T as the end time and deltaT the time scale.

The relevance of this approach will not lie in a generally usable method for coupled simulations, but it seems to be an efficient solution for scenario analysis considerations as explained in section 2. In the context of the project, the scenario analysis implies two subtasks: Firstly, the description of data structures to define initial settings and external influences on the setting during the simulation period for a single scenario. Secondly, the set of scenarios that has to be calculated. The definition of the experiments will be oriented according to the deliberations of Wittmann in [9] where an elaborated specification for experiment specifications is undertaken. Giving such an initial description what to do during the scenario calculations and with the knowledge which of these runs contain feedback structures, because those feedbacks have to be defined therein, the iteratively driven simulation can be applied very precise and only for those cases it will be necessary. With such a sophisticated application the disadvantage of extremely long simulation times is reduced on a small number of cases the method has to be used and the advantage of easy model architecture with independently definable and autarkic submodels counterbalances the effect of the prolongation of the simulations. Together with an explicit experimental specification oriented on the work of Wittmann in [9], a clearly defined solution will be built.

5 Conclusions

The initial task was generated by the project "efficient airport 2030" establishing a model architecture to simulate the passengers way from the home through the airport up to the take-off of the aircraft. Analysis showed that the main process is a stream of passengers that can be modeled by a pipeline of submodels in a model chain. With this model pipeline concept the black-box character of the submodels can be assured, a fact that was very important in the context of the project because the submodels worked with completely different model scales and with different model description paradigms. Defining the scenarios it becomes evident that this architecture does not fulfill all the demands because it cannot handle any feedback processes in the pipeline. For restricted situations and a raw scale the concept of iteratively driven simulation is proposed.

Before the advantages of this concept shall be concluded it should be pointed out that there are intrinsic restrictions using this approach. There is no feedback possible within a simulation interval deltaT even with the iteratively driven simulation. Theoretically, the time simulation interval could be reduced to an arbitrarily small size, but this would increase the simulation runtime substantially. This effect is observably at other concepts for model coupling, too, but there better runtimes are achieved by much more effort in synchronization and communication between the submodels.

Special treatment has to be done concerning the modeling of random processes. There would be problems if these random processes would be realized as real random processes and not –as usual in simulations- as reproducible pseudo random processes. If the submodel would be able to deliver different results in a re-simulation the concept of the feedback mechanism would cause inconsistencies.

As a special advantage the autarky of the submodels has been elaborated. From the viewpoint of the experiment execution, the submodels can be regarded as completely stateless, needing as initial information for a simulation run only the initial setting and no further intermediate restarting points. As the submodels the experimentation environment is stateless, too, because basically, there is only the functionality to initialize and to start a submodel. The decision about the relevance of a model situation for feedback has to be modeled explicitly within the submodels and has to be put to the output set of information of the respective submodel.

The concept is in prototypical use in an implementation with web services [11] and the XML Pipeline Language (XPL) [12]. First benchmarks suggest the transferability on other applications holding the precondition of an only loose feedback coupling.

6 Literatur

[1] Bach, M.; Himstedt, K.; Wittmann, J. (2003): Einbindung von Simulationen in computergestützte Lernumgebungen auf der Basis einer Client/Server-Architektur. In: Wittmann, J.; Maretis, D. K. (Hrsg.): Simulation in Umwelt- und Geowissenschaften. Workshop Osnabrück 2003. Shaker Verlag. Aachen (2003) 109-128.

[2] Gausemeier, J.; Pfänder, T.; Wenzelmann, C. (2009): Strategische Unternehmensführung mit Szenario-Management. In: Bullinger, H.-J.; Spath, D.; Warnecke, H.-J.; Westkämper, E. (Hrsg.): Handbuch Unternehmensorganisation: Strategien, Planung, Umsetzung. Springer-Verlag. Berlin.

[3] Phleps, P. (2009): Airport 2030 – Szenarioerstellung im Rahmen des Leuchtturmprojektes „Effizienter Flughafen". Studie im Auftrag des DLR Institutes für Lufttransporte und Technologiebewertung. Lehrstuhl für Luftfahrttechnik. Technische Universität München.

[4] Raab, M.; Schulze, T.; Straßburger, S. (2008): Erfahrungen aus der Anwendung von HLA-basierter verteilter Simulation im Nutzfahrzeugbereich. In: Rabe, M. (Hrsg.): Proceedings der 13. ASIM-Fachtagung Simulation in Produktion und Logistik. Fraunhofer IRB Verlag. Stuttgart (2008) 499-508.

[5] Rohwer, J. (2004): Konzeption und Implementierung eines Modellservers zum browserbasierten Zugriff auf Simulationsmodelle. Diplomarbeit. Arbeitsbereich Technische Informatiksysteme (TIS) am Department Informatik der Universität Hamburg.

[6] Spitzencluster Hamburg (2009): http://luftfahrtstandort-hamburg.de/index.php?id=576 , Stand 25.08.09

[7] Straßburger, S. (2006): The Road to COTS-Interoperability: From Generic HLA-Interfaces Towards Plug-And-Play Capabilities. In: Perrone, L. F.; Wieland, F. P.; Liu, J.; Lawson, B. G.; Nicol, D. M.; Fujimoto, R. M. (Hrsg.): Proceedings of the 2006 Winter Simulation Conference. Monterey (2006) 1111-1118.

[8] Straßburger, S.; Schulze, T.; Lemessi, M. (2007): Applying CSPI Reference Models for Factory Planning. In: Henderson, S. G.; Biller, B.; Hsieh, M.-H.; Shortle, J.; Tew, J. D.; Barton, R. R. (Hrsg.): Proceedings of the 2007 Winter Simulation Conference. Washington, D.C. (2007) 603-609.

[9] Wittmann, J. (1993): Eine Benutzerschnittstelle für die Durchführung von Simulationsexperimenten – Entwurf und Implementierung der Experimentierumgebung für das Simulationssystem SIMPLEX II. Dissertation. Friedrich-Alexander-Universität. Erlangen-Nürnberg.

[10] Wittmann, J.; Himstedt, K.; Möller, D. P. F. (2009): Ein Pipeliningkonzept zur Modellierung der Passagier- und Gepäckbearbeitung am Flughafen. In: Gnauck, A.; Luther, B. (Hrsg.): ASIM 2009 – 20. Symposium Simulationstechnik in Cottbus. Shaker Verlag. Aachen (2009) 404-414.

[11] W3C (2010a): Web Services Activity; http://www.w3.org/2002/ws , Stand 05.03.10

[12] W3C (2010b): XML Pipeline Language; XPL Version 1.0 (Draft). http://www.w3.org/Submission/xpl , Stand 08.03.10