Business Process simulation with the cloud-based Massive Multi-Agent System MARS

Christian Hüning, Eike-Christian Ramcke, Stefan Sarstedt, Ulrike Steffens

Abstract

Analyzing and understanding large-scale, distributed business applications and enterprise architectures is particularly important to align an enterprise's overall business with the supporting IT infrastructure. It is necessary to model and simulate the dynamic complexity properly as well as capture possible emergent effects in real-world enterprise architectures and process landscapes. So technologies and methods that extend static analyses, like delivered by typical enterprise architecture tools, are needed.

This work proposes an approach for such dynamic analyses in which agent based modeling and simulation is utilized through the MARS System, a cloud-based massive multi-agent system. The system continuously simulates the operation and transformation of the application landscape in order to support advancing business scenarios. Agents in this approach may represent any entity. They may be customers and suppliers as well as users and stakeholders of the business application, but may also represent any internal component of the enterprise system like servers, services or certain applications.

MARS is an MSaaS (Modeling & Simulation as a Service) system, which enables users to setup and control complex, distributed massive multi-agent simulations from within a web application. Models can make use of GIS data and integrate it with arbitrary information from other sources, thus creating a multi-layer environment which depicts the whole business application landscape. This paper presents the current state and possibilities of the MARS system and discusses an example model to showcase the agent-based simulation approach and its possible advantages for dynamic and continuous enterprise architecture analysis.

1 Introduction

Analyzing and understanding large-scale, distributed business applications and enterprise architectures is particularly important to align an enterprise’s overall business with the supporting IT infrastructure. It is necessary to model and simulate the dynamic complexity properly as well as capture possible emergent effects in real-world enterprise architectures and process landscapes. So technologies and methods that extend static analyses, like delivered by typical enterprise architecture tools, are needed.

This work proposes an approach for such dynamic analyses in which agent-based modeling and simulation is utilized through the MARS System ([HWFT14]), a cloud-based
massive multi-agent system. The system continuously simulates the operation and transformation of the application landscape in order to support advancing business scenarios. Agents in this approach may represent any entity. They may be customers and suppliers as well as users and stakeholders of the business application, but may also represent any internal component of the enterprise system like servers, services or certain applications.

An initial step to enable a simulation-based dynamic enterprise architecture analysis is the mapping of typical enterprise architecture meta model entities to a MARS simulation model and the simulation of some first simple enterprise architecture scenarios in order to validate the approach of dynamic analysis. This paper sketches concepts and exemplary scenarios in order to achieve this. It has to be understood as an initial research outline. Described scenarios currently focus the more technical levels of enterprise architecture. Future research will consider business levels more intensely.

The paper is organized as follows: Chapter 2 provides an overview of related work and connects our proposal to it. Chapter 3 outlines our general concept, further details how simulation models are built with MARS and more specifically how enterprise architecture can be mapped to that approach. Chapter 4 presents the example model ‘HAWAICYCLE’ including scenarios for the simulation. Finally chapter 5 sums up possibilities and advantages of our approach and chapter 6 provides an outlook to future work.

2 Related work

[BFKW06] provide a systemic overview of enterprise analysis techniques and substantiates their application by an exploratory study. [BuMS09] present a classification schema for enterprise architecture analysis approaches. They identify dynamic behavior as an interesting object of analysis, particularly if pathological effects like system failures are considered.
Recent work published by KTH Royal Institute of Technology deals with the analysis of specific quality attributes based on meta-models. These models represent application landscapes and are used to calculate indicators for every application based on attributes, which are retrieved from previous measurements or educated estimations. The indicators then allow judging the whole enterprise architecture based on selected quality features. Examples for the indicators are availability ([FrJK14]), modifiability ([Lage10]), reliability ([Köni14]), and data accuracy ([Närm12]) amongst others.
[Bu++09] combine a pattern-based approach towards typical enterprise architecture analysis concerns conceived at the Technical University of Munich with the Probabilistic Relational Models used by KTH. The informational patterns described in this work form specific meta-models for the analysis and can also be used as initial source for MARS agent and environment entities.
3 Concept

The idea behind business process modeling with a multi-agent system is to have a (partially) deployed enterprise architecture, which is used by a corresponding multi-agent simulation model to simulate usage, heavy load, restructuring and outages, among other events, against that system. The goal is to gain a better overall understanding of the enterprise architecture as well as retrieve measurable indicators, which help in making educated decisions in enterprise architecture management.

Prior to simulating it is crucial to specify the problem or the question which is going to be investigated by the simulation as well as the required indicators, which need to be monitored. It must be decided which entities from the architecture will become simulation assets, that is, are translated into agents and environment.

Basically every stakeholder in enterprise architecture may be represented by an agent in the corresponding simulation model. To clarify, this applies to all levels of an enterprise architecture (c.f. [Dern09] for typical enterprise architecture levels). So agents may be customers, marketing experts and sellers or buyers, but also IT specialists like administrators and developers, or even hackers. Furthermore, it is also possible to have technical agents. These could include agents wrapping the actual software components of the landscape, while reflecting and delegating the component’s interfaces. In this way, technical agents can either gather statistical data about the usage of the component or even manipulate the data flow within the application landscape.

The environment might be different for each type of agent. An administrative agent on the technical level might learn about servers, their loads and network capacity as result of environmental exploration, while a customer agent could retrieve information about prices or place his orders.

3.1 Modeling with MARS

3.1.1 Overview
A central concept of creating a working simulation model with MARS is the layer approach. It is inspired by the way GIS files are composed. These files are structured in layers, where each layer represents a specific aspect of the depicted real world. This aspect may be an agent type as well as a part of the environment.

We translate this GIS idea to a general approach for modeling the implementation of our simulation system. A domain-specific model may be transformed into working code by writing a layer for each aspect from the model. An aspect should be a considerably sized, self-contained but yet manageable piece from the original model.
With that approach in place we can apply well understood techniques from software engineering and thus understand layers as components with interfaces in the MARS distributed simulation system.

### 3.1.2 Mapping EA models to MARS

The layer concept is a natural fit to typical enterprise architecture meta models in general and to the ArchiMate modeling paradigm in particular. ArchiMate already defines three major layers, which are utilized to structure the enterprise architecture model. These layers are the Business-, Application- and Technology Layer ([Lank08], [Open13]). Each of them hosts a well-defined subset of the overall enterprise architecture and thus provides a good starting point for the simulation model.

Depending on what should be simulated, we propose a top-down approach, which may start with three MARS layers corresponding to those in the ArchiMate model and be further refined as the model grows deeper and / or more complex and simulation tasks get more specific. Each of the layers may contain entities from the enterprise architecture model either as agents or as a form of environment, which is to be explored and used by the agents (e.g. service applications as agents, using worker nodes exposed by the environment).

As the model grows, the top-level layers should be considered more like logical layers than real layer-components. Each category may then consist of multiple MARS layers, which together form a logical representation of the corresponding enterprise architecture layer.

### 3.1.3 Linking the model to the real world

It is up to the modeler to decide where the model ends and the real world starts. If a central question of the simulation model is, how a change in customer behavior reflects in the deployed and productive enterprise system, the MARS model would likely focus on the Business Layer. In that case, agents resemble customers with a certain (changing) behavior utilizing the provided business processes. The simulation would thus target the social and economic behavior of customers, while running against a duplicated real instance of the production system.

### 4 Example model „HAWAICYCLE“

To further illustrate the proposed approach, a sample model of fictional enterprise architecture has been created by the HAWAI project group (see figure 1). The model features a yet simple architecture comprising four business applications tied together by means of REST, SOAP, Message Queues or simple file-import interfaces. It is meant to resemble a status quo of an enterprise architecture created by an uncontrolled evolutionary process, which is by no means ideal or perfect (compare [En++08]) but not
unusual within today’s enterprises. Also administrator agents might later add additional redundant instances of all applications as they see fit.

To leverage this enterprise architecture model, we want to investigate different possible scenarios:

4.1 **Outage on the Business Application or Technology Layer**

This scenario features a large amount of customer agents on the business layer which produce load by ordering products through the HAWAICYCLE web shop. A hacking agent is present in the Application or Technology layer and will randomly kill one of the business applications or services. It is ensured that only one application is taken down at any given time. The outage of that application will be noticed by one of the (few) administrator agents, which will then attempt to bring the application up again.

---

Figure 1 HAWAICYCLE Architecture
A satisfaction indicator on the customer agents will be altered in accordance with the outcome of their transactions. That is whether the system was capable of successfully processing the order or failed in doing so. It is of course also possible to utilize metrics like the duration of the request in the calculation of the satisfaction value.

Analogous to a failing business application, an outage on the technology level might be simulated as well. In this case the hacking agent shuts down a database server instance or any other backing technology, while the administrator agent will try to bring it up again. Again, customer agents will react to possible delays or terminations of their requests.

Comparing the severity of these two possible outages will yield interesting insights about the robustness of the software solutions used. It might well be that a temporarily outage of a database instance doesn’t affect the business applications at all due to in memory storage of business data and as long as the instance comes up again in a critical timeframe. A refined simulation model might take this into account, in order to validate options for improvements on the technical level of the real enterprise architecture.

4.2 Modifications on the Application Layer
This scenario handles modification of the live application landscape. It includes extending, deleting, adapting and restructuring the enterprise architecture ([Lage10]). Enterprises, at a more or less regular basis, introduce new software components into an existing enterprise architecture and also migrate existing components. This can, however, be a daunting task, since components are usually interconnected by a large number of interfaces, exchanging data in order to realize the enterprise’s business processes. To be sure, that a new application instance fits into the landscape, the enterprise architect has to make sure that it a) meets all functional requirements and consumes and provides data in a semantically correct way and that it b) also meets non-functional requirements (e.g. it should meet performance requirements given by the business processes in which it is involved). Enterprises with mature enterprise architecture management document these interfaces and the data exchanged. However, especially for Commercial off-the-Shelf Software, but also for historically grown parts of the application landscape, this is not always possible.

In a respective simulation scenario, the aforementioned technical agents can be used to simulate the data exchange between the existing landscape and the new software application. Data exchange is driven by business processes induced by customer or employee agents. A simulation could e.g. explore whether all affected process still run through or could employ monitoring in order to document opaque interfaces.

4.3 Mutations on Dependencies
This scenario features agents which may intercept data sent to a component’s interface, analyze or even change it and finally delegate it to the targeted component. To accomplish that behavior, an agent can offer the same endpoint as the original component does. Of course customer agents may also provide erroneous data in the first place.
This data alteration might either be detected by applications throughout the system or aggregate to the final receiver. The scenario is meant to highlight weaknesses in the interaction between components and their interfaces. For instance huge problems might arise if a product’s prize is too low if multiple data sources are being used.

5 Possibilities & Advantages

With our approach we hope to go beyond the realm of testing in that we are capable of creating situations which emerge from the individual variable fined-grained behavior of agents. We hope these emergent effects can challenge alternative directions in enterprise architecture evolution which otherwise would require intense and costly preparation or are nearly impossible to test in the classical way. This should apply for both, the change at IT level (e.g. application components or infrastructure) as well as at business level (e.g. business processes).

Furthermore, we propose the usage of MARS in combination with a deployed version of the system to build a continuously running and regularly used test bed for integration tests and continuous delivery of new features. Given a well-designed set of indicators, the influence of new features, different solutions or tiny tweaks in either hard- or software could be visualized and analyzed. This allows for a quick evaluation of changes to the enterprise architecture.

6 Outlook

In this paper we outlined our concept, first ideas and possibilities to model business processes with the massive multi-agent system MARS. The current state of MARS is that of a Beta Release and first serious models from various domains are being created as of the writing of this paper. MARS will be made available as an MSaaS (Modelling and Simulation as a Service) ([Cayi13], [Padi14]) system in the near future.

The HAWAI project has just kicked off development of the initial HAWAICYCLE system and its surrounding infrastructure. The infrastructure for HAWAI will be created as a “as a Service” model, which is planned to deliver the possibility to roll out new HAWAICYCLE landscapes in various configurations on the fly. This will be achieved by the means of software-defined services, virtualization technologies like LinuxKVM and Docker, as well as tools like Puppet etc.

Parallel to this, more elaborate enterprise-architecture-specific simulation models will be created over the course of this year. These will include both outage scenarios as well as one additional scenario more concerned with business aspects of enterprise architecture management.

Future research will include the further expansion of our test models, validation of the behavior of the simulations, especially in comparison with existing static enterprise
architecture analysis, as well as a coupling of the simulation environment with existing enterprise architecture tools and respective visualization concepts. A long-term goal is the creation of simulation models according to influential IT trends, such as enterprise cloud usage or the German Industrie-4.0-Initiative.

Bibliography


[En++08] Engels, Gregor; Hess, Andreas; Humm, Bernhard; Juwig, Oliver; Lohmann, Marc; Richter, Jan-Peter; Voß, Markus; Willkomm, Johannes: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten. Dpunkt Verlag, Heidelberg, 2008.

Contact

Christian Hüning
HAW Hamburg
Berliner Tor 7, 22099 Hamburg
T +49 40 42875-8436, christian.huening@haw-hamburg.de

Eike-Christian Ramcke
HAW Hamburg
Berliner Tor 7, 22099 Hamburg
eike-christian.ramcke@haw-hamburg.de

Prof. Stefan Sarstedt
HAW Hamburg
Berliner Tor 7, 22099 Hamburg
T +49 40 42875-8434, stefan.sarstedt@haw-hamburg.de

Prof. Ulrike Steffens
HAW Hamburg
Berliner Tor 7, 22099 Hamburg
T +49 40 42875-8184, ulrike.steffens@haw-hamburg.de