Aspects of Model Interaction in Mechanized Tunneling

K. Lehner[1,a], K. Erlemann[1,b], F. Hegemann[1,c], C. Koch[1,d], D. Hartmann[1,e], and M. König[1,f]

[1]Chair of Computing in Engineering, Faculty of Civil and Environmental Engineering, Ruhr-University Bochum, Universitätstr. 150, 44801 Bochum, Germany [a]PH +49-234-32-27416; FAX +49-234-32-07416; email: karlheinz.lehner@rub.de; [b]kai.erlemann.lehner@rub.de; [c]felix.hegemann@rub.de; [d]koch@inf.bi.rub.de; [e]hartus@inf.bi.rub.de; [f]koenig@inf.bi.rub.de;

ABSTRACT

Underground infrastructures are becoming increasingly important components of modern traffic concepts worldwide. This includes, in particular, investigations of the stability of tunnel faces, material models for subsoil behavior, damage analysis of tunnel linings and supports as well as process-oriented simulation models for mechanized shield driving. Due to the strong interaction between the individual tasks in mechanized tunneling, the exchange of data and the interplay of components in simulation, the Collaborative Research Center SFB 837 has been established at the Ruhr-University of Bochum. In this paper a brief overview on the 14 sub-projects in the SFB 837 addressing one of the characteristic research areas in underground engineering is given. The main part of the paper deals with key sub-projects solely dedicated to the computer-supported integration, visualization and interaction of related models and information.

INTRODUCTION

The focus of research of the Collaborative Research Center SFB 837 “Model Interaction in Mechanized Tunneling” started at the Ruhr-University of Bochum in 2010 is placed on two main issues. First, there are several sub-projects concerned with fundamental research problems regarding specific aspects of mechanized tunneling. This includes subjects such as

- recognizing subsoil structures based on the analysis of machine data and creating material models for destructuring subsoil behavior,
- using acoustic techniques for underground exploration,
- investigating the stability of tunnel faces,
- creating process oriented simulation models for mechanized shield driving, including monitoring-based optimization of process work flows and
- employing methods of system identification used for the adaption of numerical simulation models.

Second, as can be seen from the above list of highly interrelated tasks, a further focus of research addresses the question of how the individual project models have to be coupled and how data and ideas can be exchanged in an efficient, collaborative and practical manner to create synergetic effects that will notably increase the productivity and creativeness as a whole. In addition, it is of interest how designers, engineers, managers, TBM operators, maintenance workers and others can successfully collaborate during the actual construction phase, using tools and ideas developed within the research projects.

Thus, a specific sub-project (D1) is in charge of the implementation of an "interaction platform in mechanized tunneling". Accordingly, this sub-project is responsible not only for the definition of purely technological aspects of interaction, such as specifying the type of a network protocol or other communication paradigms, but also for the establishment of soft skills to classify the amount and type of interaction needed. The need of collaboration was one of the important lessons learned from a similar, preceding tunneling project (TunConstruct 2010, Lehner et al. 2007, Beer 2009). In a networked environment of cooperating researchers it is therefore vital to find a proper balance between technological issues and subject-specific aspects.

BACKGROUND

System and Process Modeling. A concise and formal description of a complex system composed of interacting individual subsystems, including the rationale for phenomena created by the system, requires a systems theoretical basis that reaches into cognitive science. Based on proper tools for systems analysis, the states of a physical system can be adequately described and, more importantly, possible improvements and refinements can be found and implemented.

The problems and difficulties that arise during system and process modeling in large scale research alliances have long been a target of computer science and computational engineering. This particularly includes aspects of the management and controllability of information masses where the quantity and complexity of information input precludes an impromptu approach to information processing.

Product data management and product data models have been introduced as a mechanism to describe the entire life cycle of a product or an entire engineering system, including all relevant geometric, material, mechanical, manufacturing or administrative traits. General engineering standards such as the ISO 10303 standard for the computer-interpretable representation and exchange of product manufacturing information (STEP), or the Industry Foundation Classes (IFC) are often used successfully in various engineering fields; however, they do not (yet) have mature counterparts in underground engineering.

Product data models therefore play an important role to capture the characteristics of real world systems and subsystems, including the underlying system processes. However, because such processes usually exchange data, information or knowledge with one another, this customarily creates a large number of interactions and interdependencies. Even though an intense research activity in the field of system integration using appropriate computer models and partial models can be observed, it is still difficult to achieve a robust and loss-free exchange of data and semantics at a high level. If proper measures to guide integration are not made available in due time, then the proper and consistent interaction between project partners can be disrupted or even endangered.

Interaction Modeling. To sensibly manage the dynamic behavior of a complex real world system, such as an underground excavation site, not only the context of the overall system must be defined, but also the interactions between the individual subsystems have to be captured. This also includes the aspects of fuzziness inherent in complex engineering systems in general, but especially prevalent in subterraneous engineering. The context of a system defines the boundary of the system and the types of actors that can manipulate the components of the system (who can do what). In this respect, a modeling language such as the Unified Modeling Language (UML), a standard in current computer science, has proven to be a powerful tool to describe system context and interactions, using the concepts of interconnected objects reacting to events and the exchange of messages to communicate as typical paradigms.

One research focus in the sub-project D1 is therefore the adaption of general modeling tools (alike UML) to formally define domain specific semantics, tasks and rules, using standards such as XML Metadata Interchange (XMI) or the Object Constraint Language (OCL). Examples from a related research project are given in (Mundani, et.al. 2006) and (Niggl, et. al. 2006). A further example is given in (Hartmann 2007).

Distributed Computing. With the availability of high-speed, reliable computer networks, interaction among communicating subsystems can be handled at local (LAN), enterprise (MAN) or world-wide (WAN) level. Whereas data exchange among two cooperating software components may possibly be transacted in a local computer network, in contrast, access to the data server of an operating tunnel boring machine relies on a non-local Internet connection. In the sub-project D1, the interoperability is based upon distributed computing taking into account the following facets

- user-oriented, optimal coordination of resources and capacities, which are provided in a decentralized and dynamic manner, often using the client-server paradigm;
- efficient and problem-oriented use of open, standard network protocols and interfaces to implement resource access and inter-process communication;
- custom-made, domain-specific software service components based on existing technologies such as web services and other Internet-enabled software.

METHODOLOGY

The realization of the proposed interaction mechanism is based on a four step methodology consisting of (1) system analysis, (2) system modeling, (3) product model implementation and (4) interaction platform implementation.

Within the first step, the domain specific interaction structure between subsystems is analyzed and interacting system processes as well as various coupling parameters are identified and classified with respect to the multi-level and multi-scale characteristics of mechanized tunneling. By means of expert interviews, the domain specific terminology and knowledge regarding system states, actions, activities and tasks are formally defined. Subsequently, the system and its inherent interactions and couplings are modeled resulting in a holistic object-oriented ontology for mechanized tunneling. This ontology developed in the second step contains distributed partial models incorporating different space and time scales. Hereby, dependencies are revealed resulting in either “strong” or “loose” coupling rules, object relations, behavior patterns, data flows, events and actors interconnections.

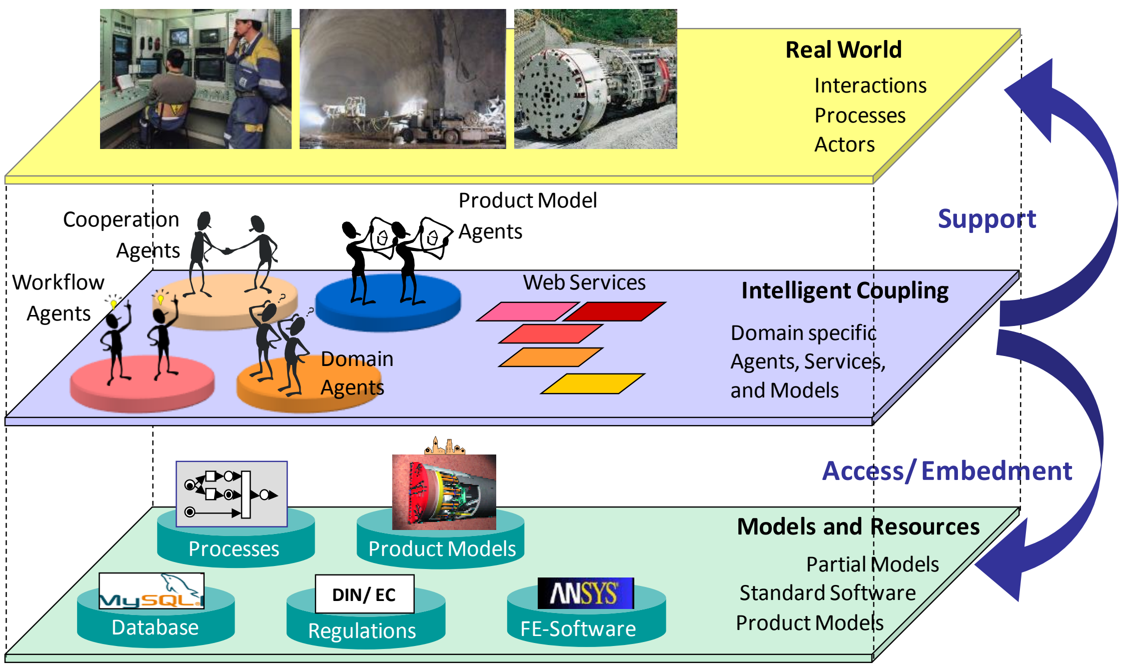

Figure 1. Three-layer Tunneling Interaction Platform (T-IP)

Within the third step, the identified components as well as the static interaction structure of the tunnel driving system are implemented as an Object-Oriented Tunneling Product Model (OOT-PM), with an emphasis on model consistency and correctness. This model is incrementally improved and enhanced within the ongoing project and provides a basis for the forth step, the Tunneling Interaction Platform (T-IP) implementation. The T-IP supports information retrieval, model updating, product and process visualization capabilities as well as a context-sensitive interaction control, in the sense of computational steering, to interactively run a holistic tunnel driving simulation. Providing a collaboration platform including system dynamics and organizational aspects, the T-IP is implemented as a three-layer architecture (see Fig. 1). On the top level, real world couplings and interactions between sub-processes and actors take place. If feasible, individual partial domain models and workflows are supported by domain agents and workflow agents, respectively (middle layer). These agents are organized in a multi-agent system, which is responsible for keeping dependencies (couplings) consistent and actually performing defined interactions between partial models. For this purpose, they access resources (bottom layer) that could be provided as Web services. If autonomous computing by means of agents is inappropriate, Web services or alternative native coupling paradigms are applied, for example High-level Architectures for distributed simulations.

Exemplary Interaction Chain. In order to test and prove the proposed methodology, the overall system complexity is first reduced to a level where a prototype implementation can be examined quickly and inconsistencies or logical flaws become immediately apparent. To this end, an exemplary interaction chain of directly interrelated subprojects has been selected and implemented. This interaction chain comprises the dependent sub-processes "Advance Exploration" and "Driving Simulation". The identified key component among these two processes is the "Ground Model", which is a dominant part of the tunneling product model (OOT-PM). An overview of the interaction workflow for the chain considered is shown in Fig. 2 and elucidated as follows.

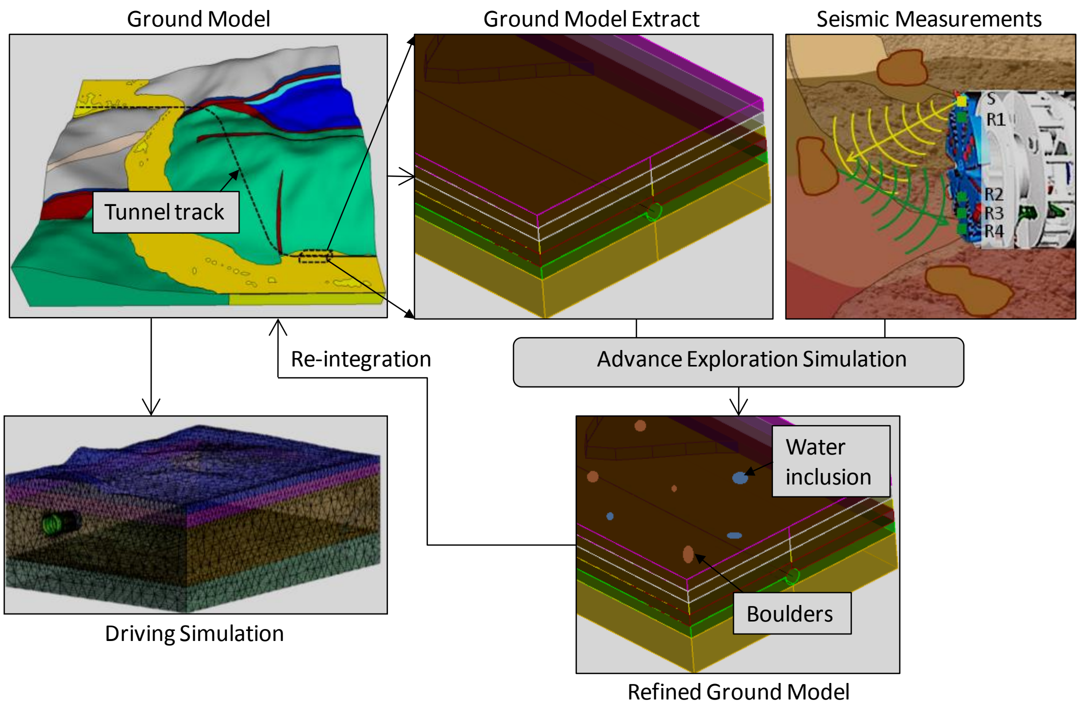

Figure 2: Workflow in an exemplary interaction chain

To perform advance exploration simulations, data from several different sources are required. On the one hand, rough geometric and geological conditions of the ground in front of the tunnel boring machine is needed to set up the simulation environment. In addition to that, current sensor data directly available at the tunnel boring machine are used as seismic measurements. At the tunnel boring machine, special actuators produce so-called preface waves, which are reflected by objects (obstacles) with a density different from their surrounding (for example boulders, water inclusions, clefts, etc. as illustrated in Fig. 2). The reflected waves are in turn detected by geophones, which measure the acceleration, velocity or displacement of the wave field. Both the ground model extract and the seismic measurements are input to the exploration simulation. Within this simulation, the sensor data is analyzed and an attempt is made to replicate the received data by changing the geological conditions in the exploration model. Through defect minimization the recognition of boulders, water inclusions or other geological irregularities is enabled and used to improve the common ground model. Then, the incrementally refined ground model provides an up-to-date basis to perform several other simulations, for example the driving simulation. Once the ground model is updated, the multi-agent system takes care of the change notification and propagation, so that other processes can benefit from the improved data set.

PROTOTYPE IMPLEMENTATION

The fundamental structure of the SFB 837, with its numerous sub-projects and participants, necessitates an adequate approach for the underlying computational infrastructure. To ensure persistency, a persistence layer is created to cope with large data sets and very heterogeneous data formats. This approach is required because, so far, no accepted common product model file format for tunneling projects exists. Furthermore, it is a fact that many subprojects are dependent on proprietary formats provided by existing simulation and analysis tools. As it is often not feasible to map all incoming simulation data files to a common objects model without loss of information, the persistence layer is used to store raw data in their respective file formats and, at the same time, to provide access to all data files as needed.

Traditional relational database management systems (RDBMS) with their rigid structures are not well suited to address the data heterogeneity needed. Therefore, a document-oriented database approach using Apache CouchDB has been chosen. In CouchDB, each document consists of a text body that uses JSON (JavaScript Object Notation) to define its contents. JSON is a light-weight text format comparable to the Extensible Markup Language (XML), but with reduced complexity and smaller computational overhead. By that, documents can be processed in the several different programming languages. Additionally, each document may have an arbitrary number of attachments, which allow to store original raw files originating from the different sub-projects. For a product model data that cannot be transformed directly into a corresponding JSON structure, the original content is stored as an attachment and annotated with a JSON document containing the meta-data necessary to find and identify the content.

As the central data repository has to be accessible from a large number of different clients, the communication layer has to provide easy access to the database content without relying on heavy-weight protocols or language-specific communication frameworks. Therefore, a RESTful approach (Representational State Transfer) has been chosen for client-server communication. REST is usually based on the HTTP protocol and allows to access and manipulate each resource by sending a standard HTTP request to a Uniform Resource Identifier (URI) that denotes the target resource. The request are processed by a Tomcat Server running dedicated Java servlets responsible for providing requested data or for updating the model.

The basic approach for processing requests based on the exemplary interaction chain is shown in Figure 3. Going back to our example, to start a new simulation run, the advance exploration client needs different sets of input data. First, it sends a HTTP GET request to obtain the geometry of all ground layers in the observed area. The file format and system boundaries are provided using URI parameters. Then, the geometry servlet responsible for processing the request fetches the relevant data from the database and transforms it into the designated target format (e.g. an ACIS file), which is suitable for generating a finite element model for simulation purposes. In a second GET request, the corresponding material parameters are fetched as JSON text and incorporated into the simulation model. Now, the simulation results can be compared to actual seismic sensor data, which are read in a final GET request. Once the simulation optimization has found an improved model, the necessary changes are send back to the respective servlets and stored in the database. As all clients have access to the same data set, all modifications are instantly accessible to other participating sub-systems (in our case. the driving simulation).

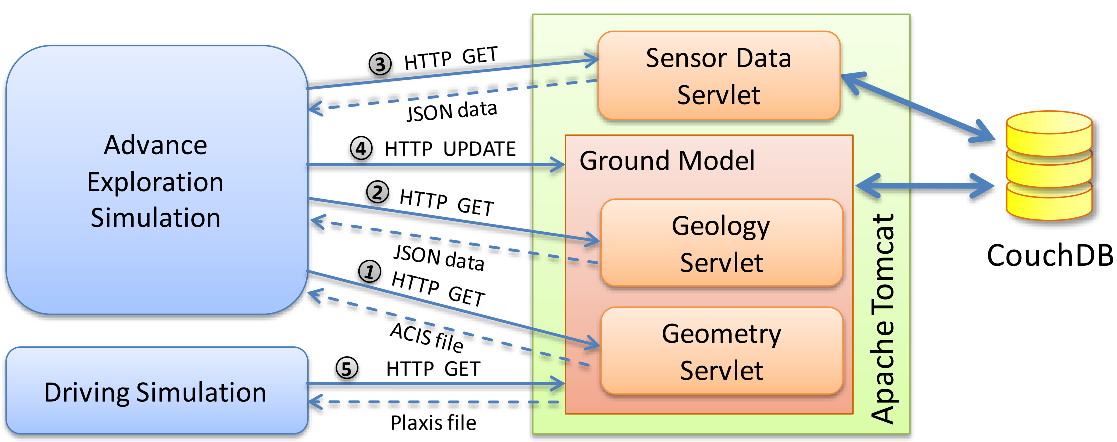

Figure 3: Service architecture and request structure for the interaction chain

CONCLUSIONS AND FUTURE WORK

The concepts, models and implementations described in this paper concerning possible interaction models in underground engineering should only be viewed as a first approach and pathfinder in the ongoing SFB 837. Although each individual research subfield in the overall project is a well-defined engineering project reflecting the state-of-the-art in mechanics, soil structures, composite materials, simulation technologies, ground modeling, etc., it is still a grand challenge problem to find the most efficient, user-friendly and technologically viable integration of co-projects to create a holistic tunneling product model and tunneling interaction platform.

The challenge itself does not only lie in the feasible implementation of technological standards and protocols, which itself can be a demanding task. Rather, because independent researchers and co-workers are actually expected to *use* the platform, subtle socio-technological aspects must be considered when designing the architecture of an interoperable system. In other words, it is not sufficient to create a large, monolith product data model and expect everyone to (re-)write their software to fit the model. Because of existing legacy systems, software components and even simple personal preferences, the driving force "interaction" implies collaboration among researchers.

Acknowledgement

The authors gratefully acknowledge the support of this project by the German Research Foundation (DFG).

REFERENCES

Beer, G. (ed.), (2009). "Technology Innovation in Underground Construction", CRC Press, ISBN 978-0-415-55105-2

Hartmann, T., Neuberg, F., Fischer, M. (2007). "Integrating 3-Dimensional Product Models into Engineering and Construction Project Information Platforms" In: Proceedings of Bringing ITC Knowledge to Work

Lehner, K., Mittrup, I., Oberste-Ufer, K., Hartmann, D. (2007). " Software Integration Aspects in a Multi-Disciplinary Research Project on Underground Engineering". In: Proc. of the 2007 ASCE International Workshop on Computing in Civil Engineering.Pittsburgh, USA.

Mundani, R.-P., Bungartz, H.-J., Niggl, A., Rank, E. (2006) "Embedding, Organisation and Control of Simulation Processes in an Octree-Based CSCW Framework" In: Proc. of the 11th Int. Conf. on Comp. in Civil and Building Engineering, Montreal, Canada

Niggl, A., Rank, E., Mundani, R.-P., Bungartz, H.-J. (2006) "A Framework for Embedded Structural Simulation" In: Proc. of the 11th Int. Conf. on Comp. in Civil and Building Engineering, Montreal, Canada.

TunConstruct (2010), "TunConstruct: Advancing the European underground construction industry through technology innovation", http://www.tunconstruct.org/