Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

A GENERIC WORKFLOW ENGINE FOR ITERATIVE, SIMULATION-BASED NON-LINEAR SYSTEM IDENTIFICATIONS

Michael Polter,
Peter Katranuschkov,
Raimar Scherer

Institute for Construction Informatics
Technische Universitaet Dresden
Dresden , 01062, GERMANY

ABSTRACT

A generic process for simulation-based system identification for civil engineering problems is presented. It is iterative and contains a feedback loop for the continuous refinement of the system, taking into account the sensor values of the real object. The process was implemented in a generic workflow engine that can be integrated into software systems and adapted to specific application scenarios. An exemplary implementation in the context of a cyber-physical system for dynamic adaptation of production processes shows the feasibility and the advantages of the developed concept.

1 INTRODUCTION

System identifications form the basis for predictions about the system and object behavior in almost all technical areas. Even if they are adapted to different scenarios, they are based on a uniform process structure: measurement of input and output variables of the system, design of a model structure with variable properties, determination of parameter values for the definition of variants, simulation-based determination of the accuracy of the modeled system using measured and calculated values. In order to map this process to software, a generic workflow engine was developed that can be adapted to certain application areas and provides hardware and software resources to efficiently carry out iterative system identification with a large number of variables.

In civil engineering, system identifications are mainly used to simulate the behavior of engineering systems, e.g. deformations, thermal behavior or fluid dynamics. They are based on simulations and are characterized by complex models with high memory consumption, resource-intensive numerical analysis processes and the need for special data structures. Due to the constantly changing system state, suitable feedback mechanisms between the physical and the digital object representation are necessary for the modeling and monitoring of construction processes. These are implemented in the process by integrating live sensors from the construction site.

A promising approach to meet these requirements are Cyber Physical Systems (CPS), which connect real and digital objects with each other and enable their continuous automatic synchronization. For the case study presented in this document, we have integrated our workflow engine into a newly developed CPS for dynamic production adaptation, in which the command and data flow between software modules from third-party providers and sensors is controlled. Access to cloud resources and data management in the form of a digital twin enable the efficient execution of iterative, simulation-based system identification tasks.

2 RESEARCH FIELDS IN THE AREA OF SYSTEM IDENTIFICATION

Nonlinear system identification is a broad field of research, with the focus shifting over the past 10 years to the industrial requirements of tools that can work with complex nonlinearities in large structures (Noël et al. 2017). In contrast to linear systems, the superpisition principle cannot be used to determine the system variables, since there are no linear dependencies between input and output variables and therefore the inverse problem can no longer be solved analytically by simply inverting the equations. Corresponding applications have developed highly specialized algorithms that are optimized for certain applications due to their complexity. Some representative research results from the very broad and lively research area are presented below.

Lafontaine et al. (2018) have developed a method for generating a dynamic system model, the input variables of which consist of pseudo-random signals that are generated using recorded signals. The output variables of the generated system model are compared with training data that have already been stored in order to determine whether they already allow conclusions to be drawn about characteristic features of the system.

Feldman and Braun (2017) deal with the identification of nonlinear vibration systems on the basis of measurement signals for free and forced vibration states. Their focus was on determining a suitable system identification procedure using the Hilbert transformation. They can identify the nonlinear parameters fairly accurately, while comparable methods only provide approximations.

Jiang et al. (2017) developed a model for the compensation of fuzzy information and measurement uncertainties. They took a reverse approach by using data stochastically generated by neural networks as the starting point for refining the system. The system then enables precise behavior predictions and independently compensates for inaccuracies in the input parameters.

Combined algorithms for transforming a time-variable nonlinear model into a time-invariant regression problem are described by Li et al. (2018). In contrast to many areas of application, which aim at the most detailed modeling of the system to be examined, the method developed here aims at the simplest possible mapping of the real system in order to reduce the parameters to be taken into account. According to the authors, the developed approach improves the traceability of parameter changes not only in the numerical model, but also in the measured input parameters of the system.

Annergren et al. (2017) have developed a framework for an optimal, application-oriented design of the input model. In particular, they also take into account the costs of creating non-linear systems, the costs being any performance parameters relevant to the user (e.g. execution time or strength of the input signal). The aim of the framework is to determine the cheapest identification method for the model in relation to the user-defined cost parameters.

Faschingbauer & Scherer (2009) first used simulation-based system identification for geotechnical structures. The aim was to continuously and in real time determine the real safety factor of the structure and to use the identification system as an early warning system to avoid a sudden breakdown, which is a recurring error in geotechnical structures.

Current research on system identification focuses heavily on the development of methods for more accurate system modeling. Processing software systems are usually highly specialized applications that are based on specific problems and thus on algorithms. We complement the research field with considerations for mapping generic, non-linear system identification methods onto adaptable software systems in order to generalize system identification approaches, to support the development of new specialized methods and to promote industrial applications.

The remainder of the paper is structured as follows: Section two is devoted to the presentation of a generic, iterative and simulation-based system identification process. In section three, we present a workflow engine that maps the general process to software and hardware and enables adaptation to specific application scenarios. Section 4 then demonstrates the evaluation of the workflow engine developed using a real construction project. Section 5 closes the paper and gives a brief outlook at the planned improvements to the developed approach.

3 SIMULATION-BASED SYSTEM IDENTIFICATION

The term system identification refers to the creation of a mathematical model of a dynamic system by identifying the dependencies between input and output parameters of an existing system, i.e. Identify the transfer function of the real system. The model created in this way shows the system with a certain accuracy and can be used to predict its behavior under certain conditions. The proposed system identification is generally done in two steps.

First, a sensitivity analysis is carried out in order to reduce the input parameters of the system to be examined. In this step, by solving a system of equations with a large number of parameter variations, it is determined which input variables have a relevant influence on the output variables. For this purpose, individual values from a large range of values are assigned to the parameters one after the other in order to achieve a significant change in the value of the output parameters in the event of dependencies in the system. In the end, by repeatedly solving the equations, all input variables can be identified that have no or no significant influence on the system behavior for the current task.

In the second step, the influence of the relevant input variables on the output variables is quantified by further simulations with more specific parameter variations. The parameter value ranges now considered are much more restricted than in the sensitivity analysis. Usually they are tailored to the current area of application and can be derived, for example, from empirical values, specified guide values or legal provisions.

In civil engineering, system identification is the basis for a large number of simulations. Figure 1 shows a schematic sketch of a system identification workflow that is abstracted from various specific applications and typical of civil engineering (Katranuschkov & Scherer 2018). The system to be determined is one of the systems of a building model, whereby the parameters considered vary. For example, these can be acting forces, thermal behavior or monetary costs. The semantic enrichment of the models, for example based on linked ontologies, enables an even better filtering of relevant system parameters as well as more precise predictions of the system behavior in subsequent simulations. However, the original models (Building Information Models - BIM) were designed by the architect from a design point of view and must first be updated to a model that is fully representative of the system under investigation, and then converted to a numerical simulation model (BIM2SIM) in which relevant system properties are modeled as variable parameters. The resulting simulation model is used as a blueprint for the generation of variants with different value assignments. Since the creation of variants is a complex process in which a large number of model candidates have to be simulated, suitable tools to support this sub-process are essential. Suitable methods for defining variants, e.g. delta encoding (Shapira 2009) help to reduce the memory resources required.

Reference values for input variables are determined by the measured or monitored behavior of the system to be examined. To automate these processes, cyber-physical systems (CPS) are required, which couple the digital representation of the system with its physical counterpart. The automated monitoring of the real object, for example based on sensors, enables a comparison of the physical and simulation-based system states almost in real time (result comparison). The final evaluation of the calculated system behavior is task-specific and can, for example, be based on legal reference specifications, empirical values or self-defined goals. If the deviation between the virtual and the measured physical system behavior is still too large, the value assignments of the input variables or the modeling of the system must be refined accordingly and the parameter studies repeated iteratively until a satisfactory match is achieved. A general reference implementation of the workflow in a special workflow engine is presented in the following section.

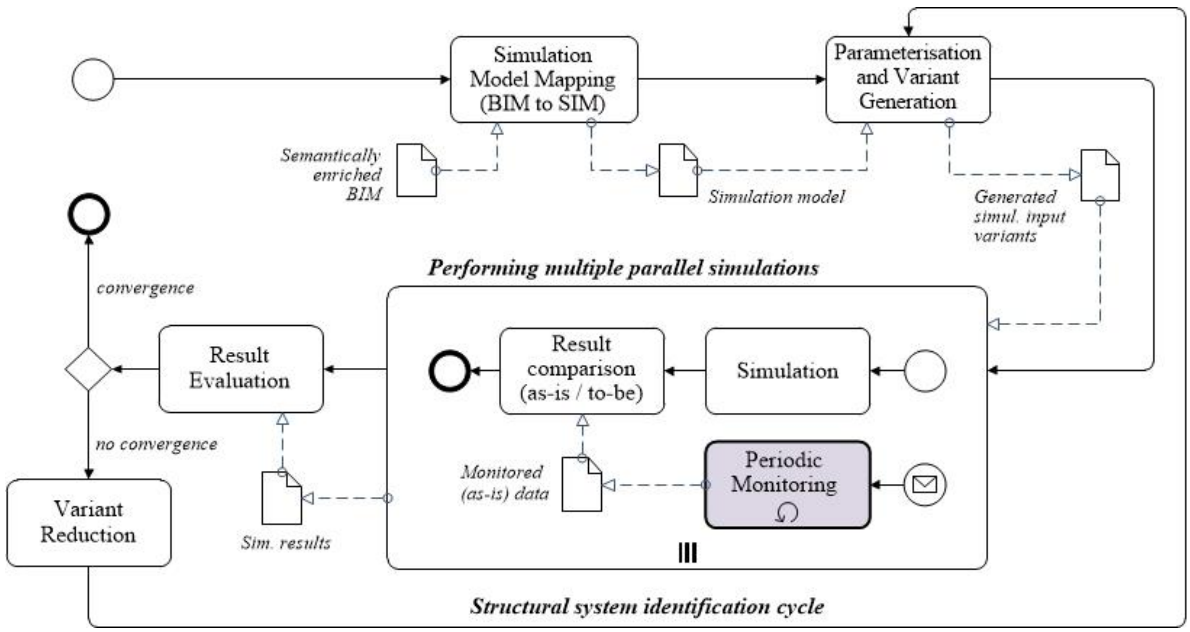

Figure 1: Generic simulation-based system identification process (Katranuschkov and Scherer 2018)

4 GENERIC WORKFLOW ENGINE FOR SYSTEM IDENTIFICATION TASKS

Industrial companies have set up workflows and integrated software products. Large companies can afford tailor-made software solutions, while small and medium-sized enterprises (SMEs) have a more limited budget. Proven software is therefore rarely replaced. As a result, many different products from different providers are often used in everyday work processes, which leads to a high degree of manual operation and problems with data compatibility. The latter is difficult to solve scientifically, since the data formats of the manufacturers are usually not accessible. In many cases, software manufacturers offer converters for their file formats to ensure compatibility without revealing the underlying data structures. This is remedied by standardized data formats such as the Industry Foundation Classes (IFC) for the cross-domain creation of building models. The automated software execution and data conversion not only enables the reduction of human input errors, but also the execution time of complex workflows significantly.

For this purpose, an intelligent workflow engine is being developed for the automated execution of a complex CPS workflow for simulation-based general-purpose system identification using existing, heterogeneous software and distributed hardware infrastructure. It enables the automation of processes in a variety of tasks, including design optimization or the constantly updated forecast of the spread of damage to buildings. The developed workflow engine is a tool for software developers, with which cost-effective software solutions can be created that are adapted to the specific established processes of an SME. It is not a software application that should be used directly by the engineers, but should always be viewed as part of a larger system, for example a virtual laboratory that is modularly developed by a commissioned software developer (Polter and Scherer 2018).

4.1 Workflow Definition

In practice, the work of the engineers is characterized by repetitive problems such as structural analysis, thermal or fluid analysis or simulations. If these general problems are refined, they represent the work processes of an engineering office and the elementary tasks contained therein.

The first task of workflow system development is to map the individual tasks of the real processes into abstract workflows. The workflow engine offers a predefined selection of general workflow steps from which the individual workflow is put together. Abstract workflows consist of a series of data entries, arithmetic operations, data storage and data output, whereby the individual steps can be carried out zero to n times and in different orders. Each of the abstract workflow steps outlined below includes atomic processes that can be automated:

- *Data entry processes* trigger a corresponding user dialog, in which the user is asked to upload or select certain data entries or to enter control information. These steps are usually at the beginning of a workflow or interrupt the automatic execution of the workflow accordingly. Data can be provided in the form of multimodels (Scherer and Schapke 2011) that are interpreted by the workflow engine.
- *Data storage processes* consist of the subtasks I) initializing the database, II) authentication, III) optionally converting the data into a specific format, IV) defining and executing the database process, V) checking the success and, if necessary, automatically repeating steps II) - V) in Error case. All of these individual steps can be defined abstractly and thus carried out automatically. The developer only needs to provide the database adapter that encapsulates the specific application programming interface (API) of the database used. Pre-built libraries are available for most database implementations.
- *Numerical analysis processes* include the calculation / simulation of a specific problem. Already purchased software products are used for this, which specialize in solving a specific problem. This software should now be started and executed automatically in the corresponding workflow steps. For this purpose, the developer specifies so-called execution engines, which represent a combination of available hardware (e.g. computers in the local network, cloud resources) and software installed on them. One of these execution engines is now assigned to each calculation workflow step, whereby it must be ensured that it also contains the third-party software required for the calculation. Examples of the definition of such execution modules can be found in section 3.4. However, calculation steps can also represent the aggregation, filtering or automatic evaluation of a series of calculation results. If no external software is required, the required processes are automatically distributed to the available hardware infrastructure.
- *Data output* processes provide the user with certain data according to the modeled process, either by visual representation or for downloading.

The above sub-processes can be put together to create any type of workflow. A corresponding Java class is currently being implemented, which provides a template for creating a workflow. For this purpose, objects of the ready-to-use (or self-implemented) workflow steps are created and configured using their methods (e.g. specification of the engine for calculation steps or the target database for data storage processes). The dependencies between the individual steps are then determined by specifying predecessors and successors.

4.2 Workflow Interpretation

In the next step, the abstract workflow descriptions are converted by the workflow engine into execution workflows (Figure 2). The task descriptions of the individual workflow steps are summarized in concrete, software-executable jobs, which are then processed by the corresponding components of the workflow engine. Such jobs include, for example, extracting / integrating data from / into multimodels using a multimodel engine, creating batch files for model calculation with third-party software using the appropriate adapters, and distributing the jobs and transferring the input / data to them corresponding target hardware. The support of multimodels for data transmission according to ISO 21597-1 (2020) is advantageous for the integration of external software tools. This improves the flexible integration of different domain models by linking only the required objects, i.e. integration by reference, in which the entire model is shown instead (Fuchs & Scherer 2017). This also facilitates the customization of third-party software APIs, as the entire process can access a single, continuously updated data structure. When parallel simulation process steps are accessed simultaneously on the multimodel, there are no inconsistencies, since each simulation run has its own input data (the model candidate) and generates its own output data. There is no parallel access to individual files within the multimodel.

4.3 Workflow Execution

The workflow engine executes the individual steps automatically based on the specified dependencies of the workflow steps. This execution is only interrupted if manual input by the user is required, a workflow step cannot be executed due to an error, or the user interrupts the execution manually. Dependencies between predecessor and successor jobs are taken into account via the *Observer design pattern*. The individual workflow steps are registered as observers with the predecessor. When a workflow step is completed, it announces this in the form of a special *finished* event that is registered by the subsequent steps (the observers) and triggers its processing. If a workflow step has several predecessors, the status of the other predecessors is checked whenever the finished event of one of the predecessors is registered. Execution only begins when all predecessors have been successfully completed. If several workflow steps have exactly the same predecessor, the workflow engine executes them in parallel by distributing them on the available hardware, which reduces the total processing time of the workflow. If a workflow step has several predecessors, e.g. in parallel simulations or when retrieving input data from multiple sources, the workflow can be blocked if one of these steps is not successfully completed. Parallel simulations are independent of one another, which is why an unsuccessful simulation can be cancelled after a certain time and the workflow can continue normally if at least one simulation was successful. For workflow steps that depend on multiple data sources, all predecessors must be successfully executed. If possible, automated error handling strategies are used here, e.g. repeated data query or re-initialization of the data source. For example, if physical sensors are damaged, an alarm informs the user that the workflow can only continue after the cause of the error has been remedied manually.

Figure 2 shows the conversion of a typical system identification process described in abstract form into an abstract workflow specification, which then becomes a concrete execution workflow by configuring the individual steps (by assigning the respective execution engine). An abstract process step can be assigned to several concrete workflow steps. For example, if data is generated by a process step, a data storage action is performed after the assigned calculation step to retain the generated data.

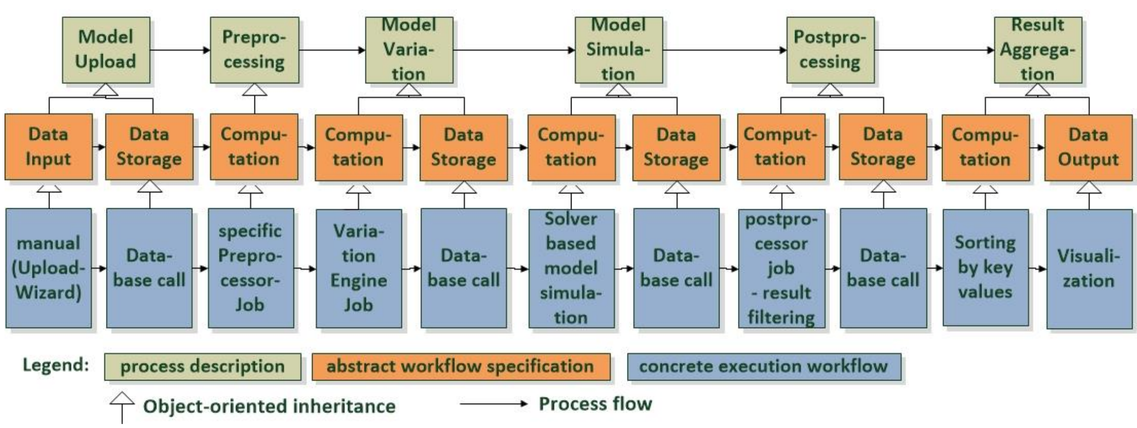

Figure 2: Conversion of an abstract process description into an executable workflow

4.4 Integration of third party software and hardware

External software tools are run by the engine through batch files that contain the appropriate command line commands. These are generated in special adapter components that encapsulate the software's command line API. The batch files are then transferred together with the input files to the target hardware of the selected engine and executed there. Distributed, heterogeneous hardware is controlled via integrated middleware, which is also controlled via adapters. Necessary file conversions are carried out automatically via appropriate converters, provided that they have been implemented by third parties or by the engineer himself.

Both software and hardware adapter components are process and task independent and can therefore be reused in various workflows. The prototype of our workflow engine is supplied with ready-made adapter components for:

- Hardware
  - UNICORE middleware for a private grid of local computers
  - jClouds-based adapter for using public cloud (currently configured for Amazon Web Services)
  - Atena Cloud platform for performing structural analyses with its own cloud access
- Solver software kernels
  - Sofistik
  - Atena
- Database
  - Hyper SQL DB as an adapter for a relational database implementation
  - ICDD framework with functions for managing multimodels

Resources are assigned to each workflow step in the form of abstract *execution engines* (Figure 3). An execution engine for simulation-based workflow steps defines a certain software and a certain hardware type for the execution of the task. Hardware types can, for example, be a specific server, but also more abstract constructs such as public cloud or private grid resources and sensor interfaces. In this way, the developer can choose to keep confidential data only on the company's network or reserve certain resources for dedicated operations. In addition, the respective network topology can be taken into account and data-intensive processes can preferably be carried out on hardware resources in the local network with a higher bandwidth. For example, an execution engine called *Atena Cloud Simulation Engine* defines a group of cloud nodes on which the Atena solver for structural analysis is installed. If this execution engine is assigned to a simulation workflow step, the workflow engine automatically distributes the simulation jobs to free cloud nodes and ensures ideal load balancing. Syntax checking and post-processing are carried out in the local company network (*Local Atena Engine*) so that the required models do not have to be transmitted over the Internet. The generation of model candidates can be resource-intensive, which is why a special HPC server (High Performance Computing) is reserved for this task (*HPC SARA Model Generation Engine*). Execution engines are defined by the platform administrator when setting up the platform. However, like workflow steps, they can also be added or updated later. This approach enables the free definition of partially automated workflows using third-party software, as long as it supports execution from the Microsoft Windows command line. Depending on the situation, the corresponding execution commands are generated dynamically by the workflow engine. It is also possible to integrate physical sensors into the workflow in real time.

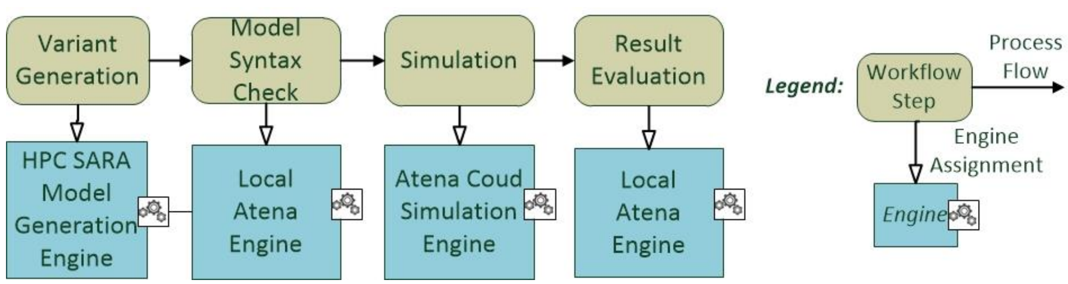

Figure 3: Assignment of execution engines to workflow steps

The workflow engine is implemented in Java and is therefore independent of the operating system. The adapter templates are currently designed for Microsoft Windows, since the predominant IT infrastructure (information technology) in SMEs runs this operating system and can therefore be used as a distributed calculation backend for the workflow engine. Migration to UNIX is possible by exchanging the adapter components. The next section contains a case study on the integration of the workflow engine into a cyber-physical system and shows the adaptability to a specific application scenario.

5 CASE STUDY: GEOPRODUCTION 4.0 PROJECT

The integration of the workflow engine into a collaborative cyber-physical system was demonstrated as part of the GeoProduction 4.0 research project. The aim of the two-year research project was to develop a web-based cyber-physical system for the dynamic adaptation of production when building all types of basic structures such as deep construction pits, tunnels, underground pipes or culverts. The production adjustment is based on forecasts using the digital twin of the basic structure (Section 4.2), which is continuously updated and refined by the construction site using the simulation-based system identification and the sensor values (Polter et al. 2020).

The platform is service-oriented and consists of several individual platforms and thus reflects the usual situation of daily practice in civil engineering with unique complex projects and always new and unique consortia. The workflow engine coordinates the control and data flow between the individual modules and thus enables the automated execution of complex processes. Individual services and entire modules can be exchanged through clearly defined interfaces. The following individual modules are part of the current implementation (see Figure 4):

- *Atena Cloud* (http://vtls.cervenka.cz/): Provision of the engine for the generation of model variants, the solver for best-fit of simulations to observations and the connection to cloud resources for parallel execution of a large number of simulations.
- *iGM.NET* (https://igm.intermetric.de/): Provision of a dashboard for remote management of the sensors and an interface for reading the sensor data directly from the construction site.
- *INERPROJECT* (https://www.zpp.de/zpp-interproject.html): Project management platform for the provision of individual models, project management data and the aggregation of sensor data from multiple sources in a uniform representation.
- *FBGuard* (http://www.safibra.cz/en/fbguard-interrogation-unit): Fiber optic sensor system with online connection for strain measurements.
- *Construction Simulation Toolkit* (Ismail et al. 2014): Construction process simulation tool. This tool is desktop-based and is required for the last step of the workflow, the construction process simulation.
- *BIMgrid* (Polter and Scherer 2018): Workflow management engine for management of the digital twin and provision of the GeoProduction dashboard.

The following section is devoted to the general description of the workflow and the adaptation of the process for identifying the abstract system to the GeoProduction system.

5.1 GeoProduction Simulation Workflow

The aim is to optimize the geotechnical structure and the associated construction process based on the knowledge gained from monitoring the actual behavior of the soil system. Therefore it consists of two parts. In the first part, the load-bearing capacity of the real floor system is identified, i.e. it is refined and its behavior examined under the conditions currently prevailing on the construction site. The knowledge gained in this way enables predictions to be made about system behavior under changed conditions. In the second part, these predictions are used to reduce the geotechnical structure system. Again, simulations are used, but now to optimize the production process system.

In this application, the system describes the behavior of the basic structure. Based on the simulation, parameter studies are carried out in order to determine the real soil parameters that cannot be adequately determined by measurements. The system identification is reversed here. The starting point are the values of the output variables (soil deformation, acting static forces) determined with the help of sensors. The aim

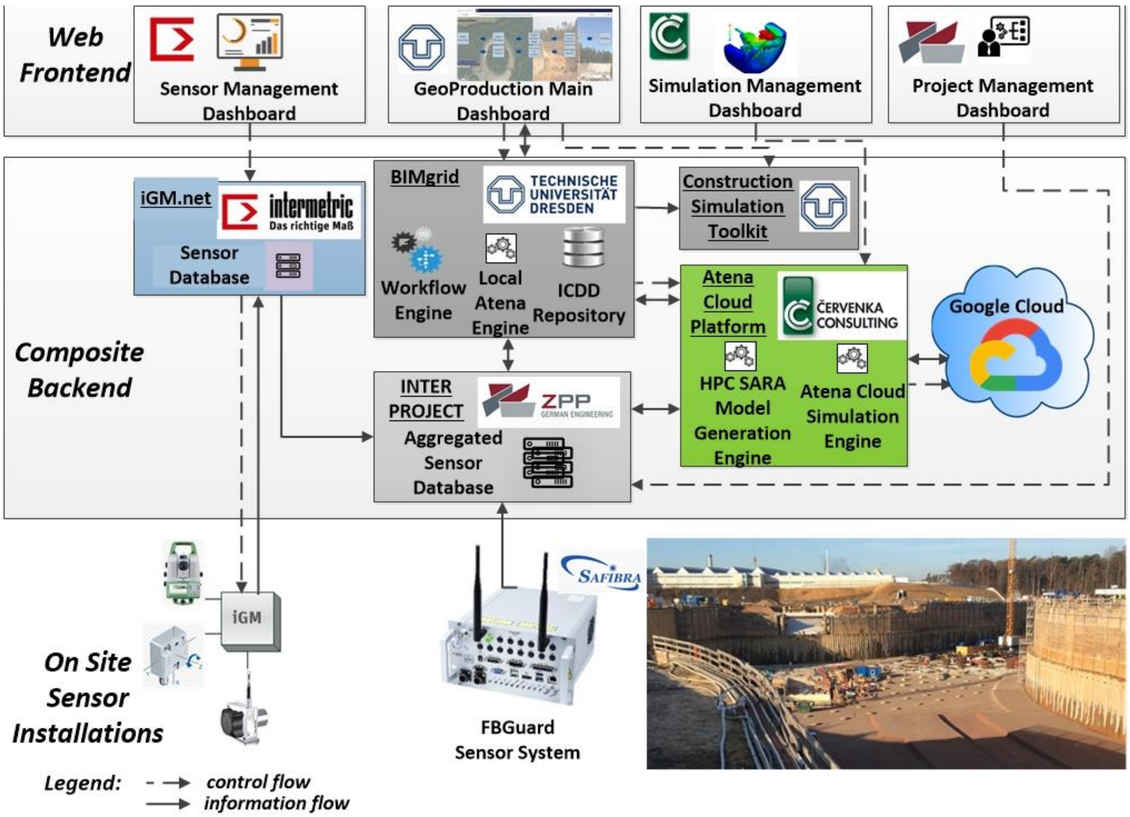

Figure 4: Top-level architecture of the GeoProduction 4.0 web platform

is to determine the values of the input variables as precisely as possible. These are represented by corresponding soil parameters that represent material laws. If these are known with sufficient accuracy, the respective model candidate realistically represents the structural-mechanical behavior of the foundation, so that it can serve as a starting point for further predictive simulations of the construction. The execution of the workflow by the platform is described below (see Figure 5):

1. Sensors are installed in the excavation pit and measure forces and deformations, e.g. on pit protection. The sensor data are collected online and brought into a uniform format.
2. The engineer creates a 3D model of the construction pit, preferably a BIM model in the standardized IFC format, and uploads it to the platform.
3. The structural engineer creates the corresponding numerical structure simulation model and defines variations of the soil parameters in a single variation model. Together with the numerical structure master model, this XML-based model forms a design for the generation of a large number of model candidates for the parametric study in order to finally determine the most suitable soil parameters. The variation model represents a number of model candidates using delta encoding. For each simulation, only the assignments of the different parameters are saved. This means that the memory consumption of the application can be significantly reduced.

4. The platform carries out simulations for each model candidate. This resource-intensive process is parallelized and thus accelerated through the use of cloud resources. The simulation results are then compared to the collected sensor data and a most suitable model is identified. If the correspondence between the calculated and measured deformations is satisfactory, the best-fit model forms the starting point for further simulations, and part 1 of the simulation is complete. The engineer defines limit values for deformations based on regulations and empirical values and the system is configured accordingly. If the deviation is not satisfactory, the suitable material parameters have not yet been found and an extended parameter study must be carried out. The process begins iteratively from step three with a new or adapted variation model.
5. As soon as a suitable structural behavior model of the excavation pit has been determined, various simulations can be carried out in order to optimize the construction process. For example, savings in anchor elements can be determined, different construction methods compared or resource planning optimized. However, since these simulations do not relate to system identification, but rather to the system optimization method, they are not explained in more detail in this paper. For more information on advanced design simulation processes, see Ismail et al. (2014).

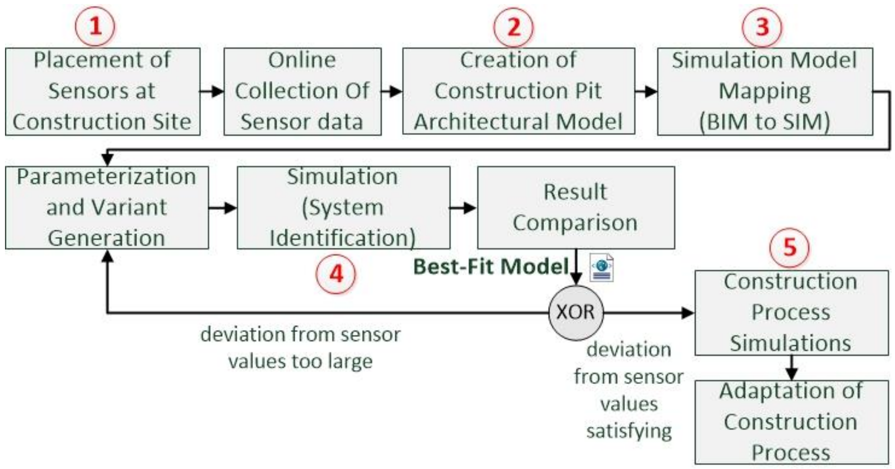

Figure 5: Simulation workflow to determine best-fit building model

5.2 Digital Twin-based simulation data management

A suitable data structure is required for the efficient execution of long-term iterative simulations. With the advent of the Internet of Things (IoT), the concept of the digital twin has established in the production industry (Jiang 2018). A digital twin describes the mapping of a material or immaterial object or process from the real world to a digital representation. In contrast to simple models, which are usually domain-specific and therefore only represent certain features of the real object, digital twins go beyond the limits of individual models. Rather, they bring together a multitude of digital object representations and information sources in order to enable as many detailed features of the physical object as possible. The connection with real-world information sources (sensors) enables a continuous synchronization of the digital object with its physical counterpart. With a digital twin, the behavior of the object can be simulated before it is actually manufactured, which is not possible with current Building Information Models (BIM) alone. The data is managed in the GeoProduction platform using ISO compliant information containers for linked document delivery (ICDD). This form of managing and linking documents, formalized processes and objects offers the necessary prerequisites for the compact mapping of the monitored structure in a digital twin with all models and data relevant to the workflow. Another argument for using ICDD as a data management format is the ability to link models or individual components with meta information that can be used for advanced filter functions, for example in a series of simulation results, in order to determine a best-fit model. The constant updating of the digital with the physical object representation is an important aspect to enable precise predictions in every phase of the object life cycle. The maximum possible frequency of the update is only limited by the speed of the underlying file system and hard drive.

The inclusion of a unique ID in the name makes the platform suitable for multiple users. For each simulation run, a folder with the ID of the simulation is created within the multimodel, which contains the simulation models stored in the delta encoding as well as the results of the static calculation and comparison values for the sensor data. According to the definition in the ICDD standard, model-based semantic links between elements of the individual models can be defined on the basis of the Ontology Web Language (OWL) (ISO, 2020). These links can also be provided with metadata and offer the possibility for software-supported semantic filter operations as well as for data mining based on artificial intelligence (AI). Data obtained in this way not only make it easier for the user to call up information, but also offer the potential for self-adapting workflows. For example, a continuous automatic aggregation of simulation results can determine the number of workflow iterations. If the sensor-supported counterpart falls below a certain threshold for the deviation of the calculated forces, the workflow execution is not restarted and the current model is presented as the best fit.

6 CONCLUSIONS

In this paper we have introduced a generic system identification process that has been specially adapted to simulation-based parameter studies in civil engineering applications. This includes the creation of the system model, the generation of the variants to be simulated, the implementation of the simulation-based system identification and the comparison with reference data from the real world. Several iterative runs are taken into account so that the accuracy of the system to be determined can be continuously increased.

Due to complex models and the large number of model candidates, this process is very resource-intensive and can only be carried out in a reasonable time with the involvement of distributed, large computing power with the possibility of parallelization and complete automation, i.e. without manual intervention. A workflow engine specially developed for this area of application addresses this through the use of distributed resources and the effective management of the simulation data in a digital twin of the system to be examined. With the GeoProduction platform, the integration of the workflow engine into a complex cyber-physical system and its extension to a digital twin was presented in a case study. The evaluation of the platform based on the data of a real construction project shows the feasibility and the advantages of the developed concepts.

As future work, we plan to expand the engine with additional adapters for legacy systems and to simplify the annotation of data with semantic information based on ready-made ontologies. In future versions, the developer and later the engineer can put workflows together using a graphical user interface. The integration of the engine into other cyber-physical systems with different application areas should underpin the concept and contribute to the further development.

ACKNOWLEDGMENTS

This research has received funding under the Grant Number 01|S17099C (“GeoProduction 4.0 – Cyber-physical controlled Production Process for Geotechnical Engineering Structures”) by Bundesministerium fuer Bildung und Forschung (BMBF) and the European Union. The funding is gratefully acknowledged.

REFERENCES

Annergren, M., et al. 2017. "Application-oriented input design in system identification: Optimal input design for control [applications of control]". *IEEE Control Systems Magazine* 37(2):31-56.

Faschingbauer, G., and R. J. Scherer. 2009. “Integrated Product and Process Model for Online Prediction and Monitoring of Geotechnical Structures”. In *Proceedings of the 16th EG-ICE International Workshop 2009*, July 15^{th} – 17^{th}, Berlin, Germany.

Feldman, M., and S. Braun. 2017. "Nonlinear vibrating system identification via Hilbert decomposition". *Mechanical Systems and Signal Processing* 84:65-96.
Fuchs, S., and R. J. Scherer. 2017. "Multimodels – Instant nD-Modelling Using Original Data". *Automation in Construction* 75:22-32.
International Standardization Organization. 2020. Information Containers for linked document delivery. https://www.iso.org/standard/74389.html, accessed 5th April.
Ismail, A., Y. Srewil, and R. J. Scherer. 2014. "Collaborative web-based simulation platform for construction project planning". In *Working Conference on Virtual Enterprises*, edited by L. M. Camarinha-Matos and H. Afsarmanesh, 471-478. Berlin: Springer.
Jiang, J.-R. 2018. "An improved cyber-physical systems architecture for Industry 4.0 smart factories". *Advances in Mechanical Engineering* 10(6):1-15.
Jiang, X., M. Sankaran, and Y. Yong. 2017. "Fuzzy stochastic neural network model for structural system identification". *Mechanical Systems and Signal Processing* 82:394-411.
Katranuschkov, P., and R. J. Scherer. 2018. "BIMification: How to create and use BIM for retrofitting". *Advanced Engineering Informatics* 38:54-66.
Lafontaine, S. R., and W. H. Ian. 2018. "Nonlinear system identification for object detection in a wireless power transfer system". U.S. Patent No. 9983243, U.S. Patent and Trademark Office, Washington, D.C.
Li, Y., et al. 2018. "Time-varying system identification using an ultra-orthogonal forward regression and multiwavelet basis functions with applications to EEG". *IEEE transactions on neural networks and learning systems* 29(7):2960-2972.
Noël, J.-P., and K. Gaëtan. 2017. "Nonlinear system identification in structural dynamics: 10 more years of progress". *Mechanical Systems and Signal Processing* 83:2-35.
Polter, M., and R. J. Scherer. 2018. "Towards the application of the BIMgrid Framework for Aged Bridge Behavior Identification". In *Proceedings of the 12th European Conference on Product & Process Modelling 2018*, September 12th-14th, Copenhagen, Denmark, 163-168.
Polter, M., P. Katranuschkov, and R. J. Scherer. 2020. "A Cyber physical System for dynamic Production Adaptation". In *Proceedings of the 14th European Conference on Product & Process Modelling*, September 2nd-4th, Moscow, Russian Federation.
Scherer, R. J., and S.-E. Schapke. 2011. "A distributed multi-model-based management information system for simulation and decision-making on construction projects". *Advanced Engineering Informatics 25*(4):582-599.
Shapira, D. 2009. "Compressed transitive delta encoding". In *2009 Data Compression Conference. IEEE*, March 16th-18th, Snowbird, USA, 203-212.

AUTHOR BIOGRAPHIES

MICHAEL POLTER is a research assistant at the Intsitute of Construction Computer Science at the Technische Universitaet Dresden. He graduated in Computer Science for Media at the Technische Universitaet Dresden. During his studies he focused on web technologies and psychological principles of user interface design. Currently he is mainly involved in the development and adaptation of hard- and software infrastructure for resource-intensive civil engineering computation problems using grid and cloud technology. His research also comprises the utilization of BIM technology for efficient data management. His email contact is Michael.Polter@tu-dresden.de.

PETER KATRANUSCHKOV, Ph.D., is an engineer with over 25 years of experience in construction IT. He has worked in a research institute and a design company in Sofia, as research asssistant at the University of Karlsruhe, and as senior engineer and consultant at Technische Universitaet Dresden where he has been the development manager in a large number of national and international joint research projects. His experience in product modelling multimodels, ontologies, distributed systems and SOA has enabled him to manage research teams successfully. His current research addresses model-based process and simulation management, system identification and digital twin development. His email contact is Peter.Katranuschkov@tu-dresden.de.

RAIMAR JOSEPH SCHERER, Ph.D., meanwhile senior full professor was head of the institute for Construction Informatics at Technische Universitaet Dresden from 1994 to 2018. He obtained his Ph.D. in stochastic earthquake engineering from Technical University of Munich in 1984 and was full professor at Karlsruhe Institute of Technology from 1987 to 1994. His research interest includes BIM, product and process modeling, multimodels, ontologies, description logic, Machine learning, virtual organizations, information logistics, digital twins, simulation based system identification, fuzzy and stochastic methods. He is the founder of the biannual European Conference on Product and Process Modeling (ECPPM) and member of the editorial board of several journals. His e-mail contact is Raimar.Scherer@tu-dresden.de.