*Proceedings of the 2014 Winter Simulation Conference*
*A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.*

# AUTOMATED PLANNING AND CREATION OF SIMULATION EXPERIMENTS WITH A DOMAIN SPECIFIC ONTOLOGY FOR SEMICONDUCTOR MANUFACTURING AMHS

Thomas Wagner
André Gellrich
Clemens Schwenke
Klaus Kabitzsch

Technische Universität Dresden
Nöthnitzer Strasse 46
D-01187 Dresden, GERMANY

Germar Schneider

Infineon Technologies Dresden GmbH
Königsbrücker Strasse 180
D-01099 Dresden, GERMANY

## ABSTRACT

To successfully manufacture logic and power semiconductors in existing high mix semiconductor factories, fast ramp up phases and frequent product changes are necessary. Especially for power semiconductor production, new manufacturing and automation concepts are required, e.g. regarding the use of other substrates than silicon wafers. To allow a judgment on how an existing automated material handling system (AMHS) can cope with the new challenges or which alterations are required, a material flow simulation is essential. However, the planning and creation of such simulation experiments is difficult because of the systems complexity, the large amount of boundary conditions and the effort of manually modifying and testing many different variants, partially using currently unforeseen automation concepts. In order to assist in this process, the authors suggest a method for rapidly creating valid simulation experiments using an ontology that allows for the reuse of previous experiments and system experts knowledge.

## 1 INTRODUCTION

Current 300mm semiconductor wafer fabrication facilities face the challenge of a diversified and fast changing chip market. Consequently, coping with fast ramp up phases as well as frequent changes of products and product mixes is the key to successful semiconductor manufacturing. An additional challenge for power semiconductor manufacturing is the use of other substrates than silicon wafers which requires new manufacturing and automation concepts. As a result, each time a new product is introduced, a production planner has to ensure that key performance indicators (KPIs) such as the cycle time or the flow factor will not be violated. But, due to problem complexity and short product life cycles the planner cannot afford the time consuming human task of manually creating detailed simulations.

Nevertheless, only detailed simulations involving the automated material handling systems (AMHS) can provide reliable insight into performance and capacity of the production, because most of the new challenges, such as increased manufacturing loads, different ramp steps, new products and new equipment, impact the AMHS significantly.

Unfortunately, manual development and configuration of detailed and valid simulation experiments in a very short time is an insurmountable task, because the following aspects of the manufacturing process have to be modeled and validated for each experiment and use case again. First, the overall system behavior has to be modeled. The core simulation logic, e.g., scheduling or dispatching algorithms, must be adapted to new products, production plans and processing tool capabilities. In high volume, high mix wafer production this often leads to higher complexity because of previously unnecessary concepts like special tool dedication or interdependencies such as products waiting for substrates. Second, transport, storage and processing have to be considered. In modern 300mm wafer production a broad variety of transport means with vastly different behavior is in use demanding new automation concepts which must be covered by the simulations depending on the particular use case. For example, overhead hoist transports (OHT) and automated guided vehicle (AGV) systems or substrate stockers and undertrack buffers as well as novel process equipment have to be evaluated before introducing them. Third, simulations have to be evaluated. To test detailed production scenarios on valid fab layouts, the simulation expert requires knowledge that usually the production experts posses, like special requirements of equipment, e.g., the availability of diverse chemicals and exhaust air units or special ambient conditions and clean room space restrictions.

As a result, it is the main objective of this paper to generate instead of manually create the required highly flexible simulations. But this can only be done if all required information about the facility, its products, process steps, control system, equipment, AMHS, requirements and boundary conditions is available in a machine-readable but human edible form. Therefore the second objective of formalizing this information as reusable concepts is derived. Hence in this paper the authors extend the previously introduced approach (Wagner et al. 2013) for generating relevant simulations by newly integrating a set of ontology modules that form a knowledge base of the semiconductor manufacturing domain. Ontologies enable automatic reasoning that can be used to assure that only valid simulations will be generated.

The remainder of this paper is structured as follows. First, in Section 2 the corresponding state of the art in the fields of modeling and simulation of transportation systems as well as production oriented ontologies are reviewed. The suggested approach of applying an ontology for generating simulations and running simulation experiments is described in Section 3. The concepts for designing a domain specific ontology for semiconductor manufacturing are discussed in Section 4 including details about mastering the mentioned challenges. The application of the approach is depicted for an exemplary use case in Section 5. Finally, in Section 6 conclusions are drawn and an outlook is given.

## 2 STATE OF THE ART

Initial layout, scheduling and routing are considered as given. Instead, the remaining freedom degrees such as arrangement of equipment and specialized auxiliary transportation systems or changing product mixes are in focus. Therefore, the relevant state of the art comprises modeling and simulation of transport systems in semiconductor manufacturing as well as ontologies for production.

### 2.1 Modeling and Simulation of Transport Systems

Simulation of transport systems has been done not as often as fab level simulations (Scholl et al. 2010) because of two reasons. First, the effort for developing a transport system simulation is very high. Second, the potential for improvements was underestimated in the past and authors just assumed certain delays between the production steps (Zhou and Rose 2012). Nevertheless, there are authors that provide case studies and surveys on design, operation and simulation of AMHS (Lin, Wang, and Wu 2003, Agrawal and Heragu 2006) but even fewer literature exists on combining both levels of simulation (Nazzal and Bodner 2003). Agrawal and Heragu (2006) provided a wide and thorough survey on design, simulation and analysis of AHMS in 300mm fabs. Many analyzed approaches are used to design the layout of future transport systems Meyersdorf and Taghizadeh (1998). Some are developed to analyze and improve the operation of AMHS (Lin, Wang, and Wu 2003; Johnson et al. 2009). Because costs of developing and running these simulations are very high, many approaches avoid to develop high detail level simulations by effective reuse (Mackulak, Lawrence, and Colvin 1998) or by finding optimal levels of detail (Jimenez, Mackulak, and Fowler 2008). In summary, many of the reviewed papers state that transport times of real AMHS exhibit large variations, which can only be explained by detailed simulations and conclude that efficient simulation development calls for further investigation. Consequently, the authors of this paper developed a software framework that rapidly generates realistic detailed simulations for testing upcoming modifications easily.

### 2.2 Ontologies for Production

Unfortunately, especially for generating simulations for existing fabs, a new challenge arises if these simulations shall be proper and realistic. Knowledge about the domain of semiconductor manufacturing has to be available machine-readable so that it can be queried by the simulation generator.

Therefore, the information about production steps, equipment and the transport system as well as its parameters, attributes, specifications and rules has to be formalized in a way that is suitable for reasoning over restrictions and conflicts as well as for easy extension so that it will support performance analysis and generation of only valid simulations. All of these requirements can be met by the introducing a generic and reusable ontology for the semiconductor manufacturing domain, see Section 4.

In (Lastra 2006), Semantic Web Services based on ontologies are proposed to cope with mass customization in manufacturing. In the field of semiconductor manufacturing control (Mönch and Zimmermann 2008), (Lemaignan and Siadat 2006) and logistic systems (Merdan et al. 2008), first approaches exist to describe transport systems using ontologies. However, the current implementations only allow for the mapping of some of the necessary logistic components, like simple conveyor elements or production equipment. For that reason, the challenges subsumed in Section 4 cannot be tackled by one of these approaches alone. Therefore, they have to be combined and extended.

For the definition of the base principles and relations, some modules of the OntoCAPE ontology (Morbach, Wiesner, and Marquardt 2009) can be reused. Although designed for computer aided process engineering, it is composed of several domain-independent reusable modules describing, e.g., technical systems and their properties and decomposition, units and physical dimensions and models. All the concepts are thoroughly axiomatized and well-documented, distinguishing OntoCAPE from many so-called light-weight ontologies. For these reasons it was chosen as the main source for upper world knowledge.

## 3 APPROACH TO SIMULATION GENERATION

Figure 1 depicts the high-level approach to generate valid factory logistics simulations by gathering information an ontology. The components of the work flow are characterized briefly in the following subsections. Crucial concepts for modeling fab simulations are detailed in Section 4. Although acquiring formal knowledge poses additional challenges, it offers several benefits:

- Knowledge about classes, e.g.equipment or transport components, is easily reusable for new use cases. Furthermore it becomes evident within an enterprise or beyond, thus fueling innovation.
- Domain models can be maintained by appropriate experts. Improvements not just affect a single software system but may easily be adopted to the whole enterprise.
- Inferencing algorithms, so called reasoners, deduct entailed knowledge from explicit assertions. Hence modeling effort can be lessened and the approach is more robust to omitted statements.
- Changes at the application level can be automatically validated by querying the domain knowledge, see Section 4.4.

### 3.1 Conceptual Knowledge and Ontology

The premise to the work flow depicted in Figure 1 is the availability of formal conceptual knowledge about manufacturing systems and their simulation. Much of this can be acquired from existing ontologies, e.g., OntoCAPE. Although very comprehensible with a focus on production, some important concepts are not included in OntoCAPE or not reusable for simulation generation such as most importantly manufactured products and their production plans, plant areas, and transport systems including their simulation. Consequently, the authors formalized these concepts in separated ontology modules, see Section 4. Thus, enabling reuse of these modules, the conceptual modeling effort in future projects is reduced to simply adding details on an application-oriented level. The Web Ontology Language (OWL) 2 (Grau et al. 2008)

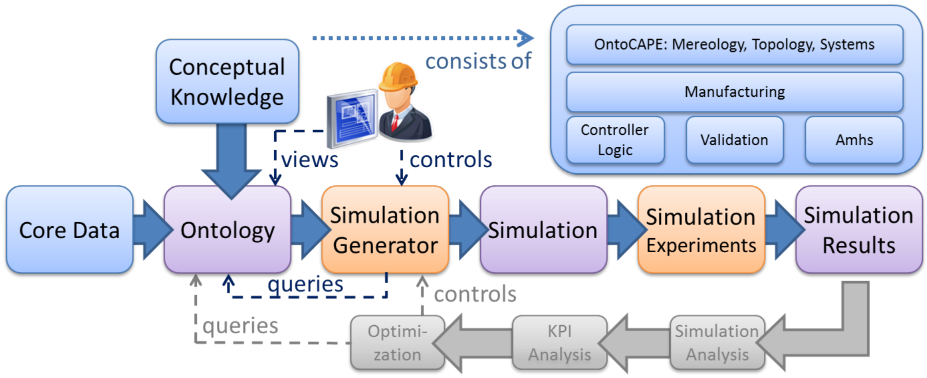


Figure 1: Approach to ontology driven simulation.

was chosen as the framing technology for processing the knowledge base because it is endorsed by the World Wide Web Consortium (W3C),it is sufficiently expressive, backed by formal logic, supported by many software tools and many ontologies are available. As a result, by the integration of existing ontologies and following their concepts the introduced knowledge base is easily extensible.

### 3.2 Core Data and Populating the Ontology

In this first step, individual fab specific manufacturing data must be mapped to abstract concepts described in the various ontologies mentioned above. This core data is usually stored in a variety of data formats and is distributed across multiple computers, servers or software services within the production plants. It contains information about, e.g., manufacturing procedures, layout, AMHS components, dispatching and routing. Since the amounts and formats, e.g., CAD-Files, CSV-Files and databases, of core data present in different manufacturing sites is not always fully known upfront, in most cases proprietary software modules are needed for this task. How to map which data to which concepts is described in Section 4. The demonstrator software tool, described in Section 5, uses a plugin-based approach to easily exchange or augment the pool of software modules for data acquisition.

### 3.3 Simulation Generator

The simulation generator component was inspired by Nazzal and Bodner (Nazzal and Bodner 2003) who suggested a complete work flow for designing AMHS, by parameterizing a template simulation based on analysis of production data. In contrast, the authors compose the simulations from small atomic templates by querying domain models from an ontology composed of conceptual and use case specific knowledge.

For this purpose a user friendly modeling front end was implemented by the authors. Using this front end, the experimenters define the variants they want to compare. For instance, the number of produced products (wafer starts per week), shall be increased. As a result, they will, e.g., rearrange or place more bottle neck tools, such as mounters, sorters, lithography tools or wet benches. The placing possibilities are automatically restricted by executing queries on the ontology. These queries are restricted to use conceptual terms on a high level of abstraction to ensure a broad range of their application presuming that all ontologies are managed by an appropriate server system running a state of the art triple store.

Querying as well as reasoning over conflicts are carried out hidden from the experimenters so that they only place equipment in the correct places. As a result a valid model of the new factory variant is created. Then the users simply hit a button and the simulation for each variant will be generated from this model. During this process atomic templates are composed that correspond to the relevant model elements which are instances in the ontology.

### 3.4 Simulation Experiments

In the exemplary use case AutoMod® models have been created. These generated simulations can be viewed, opened and changed as the experimenter pleases if necessary, but usually they will just be started. The simulations produce log files as well as instantly calculated lists of KPIs, such as cycle times, work in progress factor and tool utilization. The experimenter easily can use these KPIs to create more simulation variants using the generator until all the KPI requirements are fulfilled. To enhance the flexibility, the amount of KPIs has to be made extensible in the future with the use of software modules and appropriate concepts of the ontology. In the future this approach can be extended in order to automate this process of carrying out simulation experiments, analyzing the results and evaluating the KPIs. According to violations or improvements the necessary changes can be introduced to the factory model and the new simulations models will be generated fully automatically until sufficient KPIs are achieved. This cycle of a simulation based optimization is displayed as gray underlaid steps in 1.

## 4 CONCEPTS FOR A KNOWLEDGE BASE OF SEMICONDUCTOR MANUFACTURING

In this chapter, the major modeling concepts that support automatic generation of simulation experiments for the specific domain of manufacturing semiconductors are described. As stated above, OWL 2 syntax and semantics was chosen as foundation for the knowledge base. Accordingly, the basic modeling components are *individuals* belonging to possibly many *classes* and *properties* that manifest a specific relation between either individuals or between an individual and typed data. Predefined classes and properties can be used to describe in detail the semantics of terms, e.g., subclass relations, property ranges and domains as well as restrictions on class membership.

The IEC 62264 represents the most comprehensive body of (partially formal) manufacturing knowledge. Though it is built by domain experts, its focus is software systems integration rather than modeling the manufacturing domain. Therefore we refer to the IEC 62264 terminology where appropriate but no strict integration is attempted yet.

### 4.1 Describing Overall Production Relevant Concepts

As mentioned in Section 1, the concepts of products, product mix, process steps and process capabilities must be modeled in a way that new products with special requirements can be easily included into the simulations. In addition, to be able to validate simulation experiments created by system planners or an optimization algorithm, it must be possible to answer questions like: Does the system provide all capabilities and equipment to manufacture a planned product mix? Is a dedication requirement fulfilled to produce a special product? Is at least one equipment currently available for each step? These questions can be answered in a generic way, if appropriate algorithms are based on the conceptual model briefly described in the following.

The *Technical System* concept proposed in OntoCAPE is adapted for manufactured products and other important concepts to allow for a structured and generic approach to model product-logistic centered design aspects. This incorporates modeling the major aspects requirements, function and realization.

For the manufacturing of an individual part of a specific *Product* it is required that it follows the corresponding manufacturing plan. Consequently, individuals of the class *ManufacturingPlan* are modeled as subsystems of the requirements aspect of that particular product, as depicted in Figure 2. It is linked to required steps that identify what to do in which order, modeled by individuals of the *ProductSegment* class in adherence to IEC 62264. To allow for multiple levels of composition, they are also modeled as instances of the *ManufacturingPlan*.

Each product segment is an exclusive part of a particular manufacturing plan and cannot exist without that plan. Thus, the *isComposedOf* relation is used to link them. Furthermore it must include the relation to other product segments and act as link to a process segment. Hence the *ProductSegment* class corresponds to the concept of *SystemRequirements* that must be met and *ProcessSegment* to actual system functions that

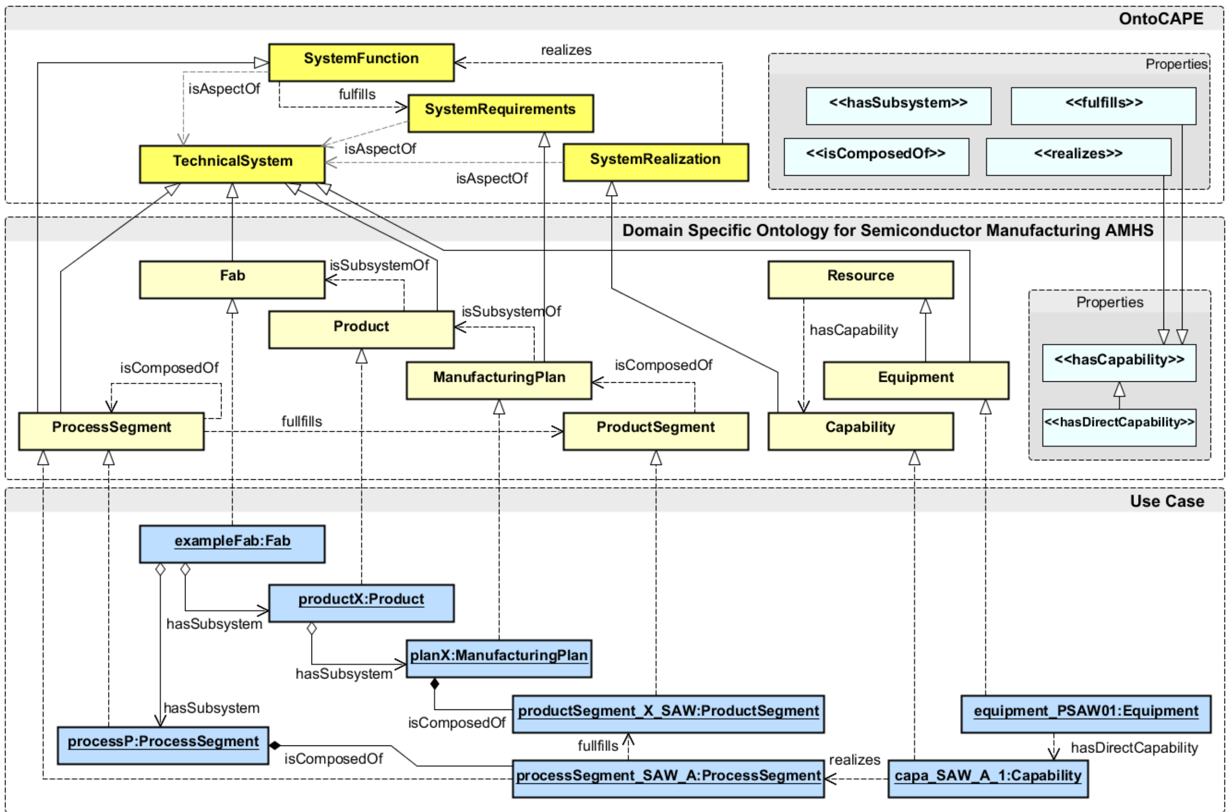


Figure 2: Abstract modeling of manufacturing plans and capabilities.

fulfill these requirements as described in OntoCAPE. Thus they should be linked with the *fulfills* relation. This can be achieved by either directly adding a connecting statement to the knowledge base or by setting up a model fragment that enables a semantic reasoner to draw the conclusion.

A specific realization of a manufacturing function (e.g. a saw process) usually incorporates a specific equipment type, other resources and appropriate process parameters. The authors model this ability of a specific *Resource*, or a set thereof, to realize a certain function as an instance of the *Capability* class and to link it to the corresponding process segment using the *realizes* property. Furthermore the capability must be linked to the corresponding resource using the *hasDirectCapability* relation. If multiple resources realize a function, they must be joined first to a composite resource which is then linked to the capability.

As described above, process capabilities only define the general applicability of resources to a manufacturing function. No conclusions are drawn about dynamic conditions at runtime of the factory or a simulation model. Nonetheless, capabilities can be considered as complex technical systems themselves. Hence, their requirements aspect offers a reasonable starting point for future extension. It should be noted that the relations *hasDirectCapability*, *realizes* and *fulfills* are subsumed by the more generic transitive *hasCapability* relation which links the abstract concepts *Resource* as domain and *Capability* as range.

All of the mentioned concepts were carefully categorized by the authors to fit the modeling paradigm of technical systems and packaged in reusable, independent modules. Furthermore, they were axiomatized with stricter axioms in the application oriented layer and more generic ones in the conceptual layer.

Given a domain model as sketched, semantic queries about products, production plans and applicable resources can be answered. As an example, the following query in Manchester Syntax and common domain terminology returns all equipment known to be suitable for the process segment *processSegment_SAW_A*.

```
Equipment that hasCapability value processSegment_SAW_A
```

### 4.2 Describing Transport System Concepts

In addition to the production related concepts described above, the general manufacturing plant structure, layout and the transport system itself must be modeled. As already mentioned in Section 1, a broad variety of different transport concepts should be covered which enables the specification of, e.g., basic transport element segments and their associated control logic like maximum capacities or process logic fragments defining routing decision. Other examples include different vehicle types regarding, e.g., capacities, allowed paths and products as well as special requirements of different AMHS types, like vehicle parking- or charging positions.

At first, a subdivision of the plant itself is modeled in accordance to the VDI Guideline 3644, which allows for various automated validation purposes, e.g. checking if manufacturing equipment is correctly placed in production areas. Then, an Ontology describing various means of relevant transport systems as well as possible use cases, e.g., 300mm OHT transport system, was developed and included in the knowledge base. The VDI Guideline 2710 presents different types of AGV systems as well as metrics on when to use them and in what situations. This information can be useful if a subsequent automation of logistic steps is considered, e.g., wafer substrate transports.

For illustration purposes, some core concepts are presented using excerpts of the modeled transport system types. The relevant excerpt of a 300mm system is shown in Figure 3, including the relevant excerpt of the resulting taxonomy starting from the class *AMHSComponent*. In the case of a 300mm manufacturing plant, the transport system has been divided into instances of the classes *StraightSegment* and *CurveSegment*, i.e., linear and curved rail sections. In addition, some of these instances are assigned to logical entities like junctions because additional logic is required to generate valid simulation models as described in the next section. This has been modeled using OntoCAPEs relation *isPartOf*. In addition, the logical relations between the components have been modeled, i.e., the predecessor and successor relations. The direct connection between two elements can be modeled using the relations *hasInput* and *hasOutput* defined in OntoCAPE. Since their abstract parent relations are transitive, it is possible to cover and deduce both direct and indirect relations between different entities. In addition, these relations can be exploited for simulation validation as outlined in Section 4.4.

As mentioned before, it is important to model the system mereology, for example to be able to gradually adjust the granularity of simulation and analysis studies. For this purpose, the relations *isDirectSubsystemOf* and *isDirectSupersystemOf* were used. In Figure 3, this is illustrated using the dashed arrows. An OWL-reasoner will for example detect, that all transport system elements that are associated to *Bay1Workcenter1* will indirectly belong to the exemplary fab bay *Bay1* by following the transitive parent relation *isSubsystemOf*.

Once the instances of the AMHS are required, they can be queried from the knowledge by simple use case independent queries on arbitrary abstraction levels. For example, the code component that generates the required process logic for a junction and then exports the logistic system model to a simulation software would perform the following query.

```
AMHSComponent that isPartOf some Junction
```

### 4.3 Describing Processing, Storage and Transport Logic

In short, transport jobs are composed of sources and sinks of routes from the transport systems perspective. In the examined use-cases, there exist two types of objects which both act as sources and sinks. On the one hand, there is productive equipment to process the goods and on the other hand there are storage facilities. Both of them are modeled as subclasses of the abstract concept *Equipment* as described in Section 4.1. Starting from these base objects that posses basic features, e.g., processing time distributions, capacities and expected availability, it must be possible to gradually differentiate specialized types of equipment with slightly different behavior as stated in Section 1. Examples include aspects like batch sizes, internal storage capacities or substrate requirements. However, more differentiated components need special process logic to be integrated while generating the simulation model.

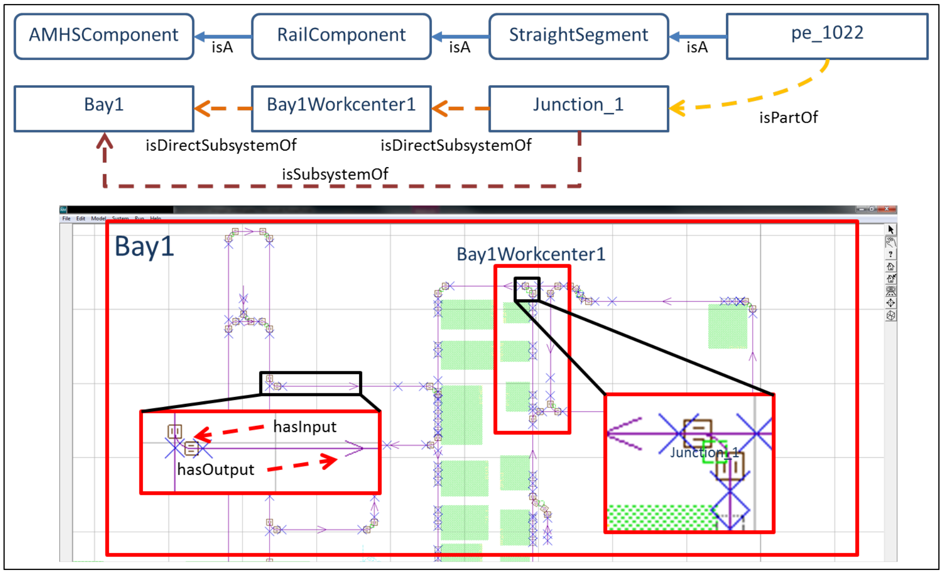


Figure 3: Excerpt of the taxonomy and mereology of a part of a 300mm OHT transport system.

For this purpose, the relation *isModeledBy* is used. For example, the generic *Equipment* class is modeled by *EquipmentLogic*, which is coupled to a generic Code component that generates stub methods or template models, depending on the simulation software used. These stubs perform basic operations, like suspending a load for a given production time or altering its runtime state. For specialized equipment types, the mentioned classes can be refined and the stub generators overridden to enhance the model with additional functionality, e.g. assembling batches or executing a detailed process chamber simulation. Similarly, the same approach is utilized for other transport system components. For instance, the process logic of a rail junction is modeled by the class *JunctionLogic* which in turn ensures that the capacity limitations of the junction are adhered to, e.g., for the simulation software AutoMod®, a sufficiently large *Block* element with capacity one is generated over the corresponding segments.

### 4.4 Knowledge Base Support for Simulation Validation

Prior to the automated simulation model generation, a validation of the layouts and parameters used as sources is necessary if they were either manually modified or automatically generated using an optimization algorithm. If correctly modeled the system is able to validate simulation experiments by answering questions like: Are all equipment correctly connected to the AMHS? Are some of the objects obstructed in their currently planned locations? Are special requirements about the location of specific equipment met, like the availability of diverse chemicals, exhaust air units or ambient conditions?

For this purpose, restricting relations as described in (Gellrich et al. 2012) and supporting validation algorithms have been created, for example the relations *hasInput*, *hasOutput* as introduced in Section 4.2. Using these relations, it is possible to define restriction about which transport system element is allowed to be connected to which others. For instance, using the following restriction it can be defined, that a straight or curve rail segment of an OHT transport system can only be connected to one input and output segment. This can easily be adjusted for merging or diverting junctions.

```
StraightSegment and (hasInput max 1 RailSegment)
             and (hasOutput max 1 RailSegment)
```

For solving more complicated validation tasks and for experiment planning, the *ModelPlausibilityCheck* and *ModelManipulation* classes are introduced. Using the latter several means of simulation model manipulations can be described, ranging from simple parameter manipulations up to larger scale changes of the production equipment and transport system layout. For validation purposes, most of these manipulations require various sanity checks which in turn are defined using subclasses of *ModelPlausibilityCheck*. For example, manipulations involving the physical rearrangement of parts of the transport system or equipment require to be checked for collisions and illegal overlapping of components. Therefore, a class collision check was defined which must be used while planning such experiments. This can be ensured by using the newly introduced restriction *requires* like

```
PhysicalModelManipulation and (requires value CollisionCheck)
```

## 5 USE CASE

Figure 4 shows the user interface of a demonstrator developed by the authors for creating AMHS simulation experiments by manipulating the parameters and the systems layout, e.g., by drag and drop of equipment. It features plugins for GUI elements, creating simulation experiments, validating their design and generating the respective simulation models as well as for user guidance by applying the concepts described in Section 4. Optional plugins for analysis, optimization and automated design of experiments will be integrated in the future.

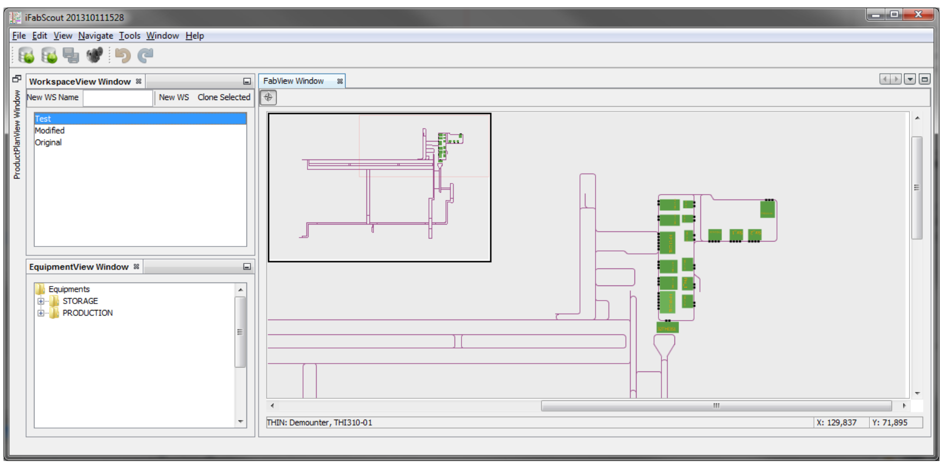


Figure 4: User interface for creating AMHS simulation experiments.

This demonstrator has been applied to exemplary industrial use cases, for example experiments for testing different production scenarios and product mixes. In one use case, a new fab production bay that is currently in an engineering phase should be tested if it can cope with future volume ramp up phases. For this purpose, first the factory core data about initial bay layout, equipment types, available and installed equipment and production steps is imported into the knowledge base as described in Section 3.2. Second, the simulation model is generated by automatically instantiating the required process equipment, rail segments and associated process logic as described in the Sections 4.2 and 4.3. After running the simulation as a third step, the KPIs can be investigated and the experimenter can improve the design by a few clicks via the GUI so that altered simulation models can be generated.

For this particular use case, new instances of bottleneck equipment were placed. This is enabled by accessing the knowledge base to validate the simulation layout in the background as outlined in Section 4.4. That is, new entities are created as instances of appropriate classes, e.g., *Equipment* in the knowledge base and queries are executed to prevent overlaps or collisions and ensuring the proper connection to the *correct* transport system rails, since in the fab at hand, distinct transport systems are used for inter-fab and intra-fab transports. In this process, only the most relevant relations are asserted, while the others are automatically deducted using the ontology.

In Table 1, the simulation results of the unmodified system are compared to the ones of the adapted bay layout. In the unmodified model, the one-of-a-kind equipment SORT01 and PSAW01 prove to be a bottleneck in the investigated ramp scenario because they are frequently used in the product plans (SORT01) or have a low production capacity (PSAW01). Table 1 not only shows the improvements of the bays throughput while using multiple instances of bottleneck tools but also proves that the new equipment instances are seamlessly integrated into the new simulation models and process logic. In addition to depicting the improvements of the time a lot has to wait for processing, TTP, a slight delay in the actual transport time from sources source to some targets, D, has to be noted due to the significant increase in transport count. It depends on other KPIs, like the tool idle times, as well as on expert knowledge whether these delays are acceptable or measures are necessary in order to reduce them, e.g., planning additional vehicles. Therefore, other plugins for KPI calculations must be included in the demonstrator and semantics about them added to the ontology.

Table 1: Comparison of the waiting times of loads to be processed at a tool (TTP, in minutes) and the average transport times of lots (D, in seconds) of the base line- and the adapted simulation model (excerpts).

| Unmodified Baseline System | | | | Adapted to Productmix | | | |
|---|---|---|---|---|---|---|---|
| Target EQ | Avg TTP | Avg D | Lot Count | Target EQ | Avg TTP | Avg D | Lot Count |
| STOR01 | 0.37 | 66.09 | 265 | STOR01 | 0.76 | 78.00 | 660 |
| | | | | PSAW02 | 0.51 | 76.05 | 34 |
| | | | | SORT02 | 0.80 | 41.78 | 224 |
| | | | | SORT03 | 0.50 | 46.56 | 301 |
| DEMT01 | 0.51 | 71.17 | 38 | DEMT01 | 0.77 | 72.46 | 107 |
| | ... | | | | ... | | |
| MOUN01 | 0.49 | 58.27 | 125 | MOUN01 | 0.49 | 58.72 | 127 |
| PSAW01 | 152.16 | 37.93 | 74 | PSAW01 | 0.44 | 77.96 | 89 |
| SORT01 | 373.85 | 37.04 | 321 | SORT01 | 0.69 | 39.11 | 247 |
| METR01 | 0.24 | 53.44 | 59 | METR01 | 0.30 | 49.56 | 116 |

## 6 CONCLUSIONS AND OUTLOOK

The authors present a methodology to assess the effects to a fab's AMHS that result from modifications, e.g., integrating new equipment or changing the product mix. The core idea is to automatically generate detailed AMHS simulations from ontological knowledge. This tremendously eases the tedious and time consuming human task of repeatedly creating and executing necessary simulation experiments. The main drawbacks are the necessity to map fab data to appropriate domain concepts and the computational complexity for simulation and automatic reasoning. While the first is compensated by the reusability of conceptual knowledge, the latter calls for massively parallel unsupervised simulation experiments. A plugin based software tool was introduced that populates the knowledge base, allows the user to freely design valid simulations and applies the knowledge for automatically generating simulation experiments.

The ontology is populated by mapping an existing fab's core data to domain specific concepts that have been formalized by the authors ensuring reusability to avoid error prone reimplementation. An extensible set of various plausibility checks is performed during the design by directly accessing the underlying ontology to derive detailed restrictions, e.g., space limitations, media requirements or necessary environmental conditions. When generating the simulations, the ontology is used again to produce robust material flow simulation models.

After having exhibited the feasibility of automated simulation experiment planning and creation, further research will focus on creating modules for automated analysis and optimization of AMHS, see gray parts in Figure 1. As mentioned above, optimizing such systems will result in a significant increase of simulation runs, thus, sophisticated methods are needed to reduce the amount of simulation experiments as well as their runtime. In contrast to estimation methods (Batarseh, Nazzal, and Wang 2010) that solve the latter problem, the introduced knowledge base can be applied to reduce the amount of simulation experiments by, e.g., ruling out non-viable layout variants. In addition, the adaption to use cases of other domains than semiconductor manufacturing will be researched, e.g., by incorporating the concepts of broader standards like the Core Manufacturing Simulation Data (CMSD) Information Model.

## ACKNOWLEDGEMENTS

This work originates in the context of the research and development projects EPT300 (Grant No. 16N12182) and EPPL (Grant No. 16ES0031) funded by the German Federal Ministry of Education and Research (BMBF) and the ENIAC Joint Undertaking (JU) and the research project Sense&React (Grant No. 314350) funded by European Union Seventh Framework Program (FP7).

## REFERENCES

Agrawal, G. K., and S. S. Heragu. 2006. "A Survey of Automated Material Handling Systems in 300-mm Semiconductor Fabs". *IEEE Transactions on Semiconductor Manufacturing* 19 (1): 112–120.

Batarseh, O. G., D. Nazzal, and Y. Wang. 2010. "An Interval-Based Metamodeling Approach to Simulate Material Handling in Semiconductor Wafer Fabs". *IEEE Transactions on Semiconductor Manufacturing* 23 (4): 527–537.

Gellrich, A., D. Lunkwitz, A. Dennert, and K. Kabitzsch. 2012. "Rule-driven manufacturing control based on ontologies". In *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*, 1–8: IEEE.

Grau, B. C., I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. 2008. "OWL 2: The next step for OWL". *Web Semantics: Science, Services and Agents on the World Wide Web* 6 (4): 309–322.

Jimenez, J. A., G. T. Mackulak, and J. W. Fowler. 2008. "Levels of Capacity and Material Handling System Modeling for Factory Integration Decision Making in Semiconductor Wafer Fabs". *IEEE Transactions on Semiconductor Manufacturing* 21 (4): 600–613.

Johnson, A., H. J. Carlo, J. A. Jimenez, D. Nazzal, and V. Lasrado. 2009. "A greedy heuristic for locating crossovers in conveyor-based AMHS in wafer fabs". In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 1667–1676. Piscataway, NJ: IEEE.

Lastra, J. 2006. "Semantic Web Services in Factory Automation: Fundamental Insights and Research Roadmap". *IEEE Transactions on Industrial Informatics* 2 (1): 1–11.

Lemaignan, S., and A. Siadat. 2006. "MASON: A proposal for an ontology of manufacturing domain". In *IEEE Workshop on Collective Intelligence and its Applications*, 6–11.

Lin, J. T., F.-K. Wang, and C.-K. Wu. 2003. "Simulation analysis of the connecting transport AMHS in a wafer fab". *IEEE Transactions on Semiconductor Manufacturing* 16 (3): 555–564.

Mackulak, G. T., F. P. Lawrence, and T. Colvin. 1998. "Effective simulation model reuse: a case study for AMHS modeling". In *Proceedings of the I998 Winter Simulation Conference*, edited by D. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, Volume 2, 979–984. Piscataway, NJ: IEEE.

Merdan, M., G. Koppensteiner, I. Hegny, and B. Favre-Bulle. 2008. "Application of an ontology in a transport domain". *2008 IEEE International Conference on Industrial Technology*:1–6.

Meyersdorf, D., and A. Taghizadeh. 1998. "Fab layout design methodology: Case of the 300 mm fabs". *Semiconductor international* 21 (8): 187–196.

Mönch, L., and J. Zimmermann. 2008. "An Ontology to Support Adaptive Agents for Complex Manufacturing Systems". In *32nd Annual IEEE International Computer Software and Applications Conference*, 531–536: IEEE.

Morbach, J., A. Wiesner, and W. Marquardt. 2009. “OntoCAPE - A (re)usable ontology for computer-aided process engineering”. *Computers & Chemical Engineering* 33 (10): 1546–1556.

Nazzal, D., and D. A. Bodner. 2003. “A simulation-based design framework for automated material handling systems in 300 mm fabrication facilities”. In *Proceedings of the 2003 Winter Simulation Conference*, edited by S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, Volume 2, 1351–1359 vol.2. Piscataway, NJ: IEEE.: IEEE.

Scholl, W., B.-P. Gan, M. L. Peh, P. Lendermann, D. Noack, O. Rose, and P. Preuss. 2010. “Towards realization of a high-fidelity simulation model for short-term horizon forecasting in wafer fabrication facilities”. In *Proceedings of the 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. R. Montoya-Torres, J. Hugan, and E. Yücesan, 2563–2574. Piscataway, NJ: IEEE.

Wagner, T., C. Schwenke, G. Schneider, and K. Kabitzsch. 2013. “Automated Planning, Execution And Evaluation Of Simulation Experiments Of Semiconductor Amhs”. In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Hill, M. Kuhl, R. Pasupathy, S.-H. Kim, and A. Tolk. Piscataway, NJ: IEEE.

Zhou, Z., and O. Rose. 2012. “Wip balance and due date control in a wafer fab with low and high volume products”. In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 1–8. Piscataway, NJ: IEEE.

## AUTHOR BIOGRAPHIES

**THOMAS WAGNER** is a PhD student at Dresden University of Technology. He is a member of the scientific staff at the Chair of Technical Information Systems. He received his M.S. degree in computer science from Dresden University of Technology. His research interests include modeling, analysis and optimization of logistic systems. His e-mail is thomas.wagner2@tu-dresden.de

**ANDRÉ GELLRICH** is a PhD student at Dresden University of Technology. He is a member of the scientific staff at the Chair of Technical Information Systems. He received his M.S. degree in computer science from Dresden University of Technology. His research interests include knowledge representation, reasoning, optimization and analysis of manufacturing processes. His email is andre.gellrich@tu-dresden.de

**CLEMENS SCHWENKE** is a PhD student at Dresden University of Technology. He is a member of the scientific staff at the Chair of Technical Information Systems at the Department of Computer Science. He received his M.S. degree in Electrical Engineering from Dresden University of Technology. His research interests include automation as well as modeling, analysis and design of discrete event systems. His e-mail is clemens.schwenke@tu-dresden.de

**GERMAR SCHNEIDER** is a manager at Infineon Technologies Dresden GmbH for “Thin Wafer Handling” and project leader for automation projects within the Eniac funding projects EPT300 and EPPL. He received a Diploma and a Ph.D. in analytical chemistry and works since 18 years in different positions at Infineon Technologies. In the recent years he filled management positions in processing, manufacturing and maintenance departments and was head of an automation and factory integration group. His e-mail is germar.schneider@infineon.com

**KLAUS KABITZSCH** is the head of the Chair of Technical Information Systems at the Dresden University of Technology. He received a Diploma and a Ph.D. in electrical Engineering and communications technology. His current projects have their focus on the automation domain, component based software design, data analysis, advanced process control and predictive technologies. He is a member of IEEE, VDE and GI. His e-mail is klaus.kabitzsch@tu-dresden.de