A General Concept for Description of Production Plants with a Concept of Cubes

Niki Popper[1,2] *, I. Hafner[1], M. Rössler[1], F. Preyser[2], B. Heinzl[3], P. Smolek[4], I. Leobner[4]

[1] dwh Simulation Services, Neustiftgasse 57-59, 1070 Vienna, Austria; **niki.popper@dwh.at*

[2] Institute of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

[3] Institute of Computer Aided Automation, Vienna University of Technology, Treitlstr. 1-3, 1040 Vienna, Austria

[4] Institute for Energy Systems and Thermodynamics, Vienna University of Technology, Getreidemarkt 9, 1060 Vienna

Simulation Notes Europe SNE 24(2), 2014, 105 - 114
DOI: 10.11128/sne.24.tn.10247
Submitted: Sept. 15, 2014 (selected ASIM SST Post-Conf. Publ.);
Revised: Oct. 25, 2014; Accepted: October 30, 2014;

Abstract. The goal of the project BaMa (Balanced Manufactoring) is to develop a simulation-based method for monitoring, predicting and optimizing energy and resource demands of manufacturing companies. Considering the economic success factors time and costs, a new modelling and simulation concept will be integrated in the research project to implement an energy and cost foot printing. A modular approach that segments a production facility into "cubes" will be developed. Cubes have a clearly defined interface and represent a certain physical behaviour that contributes to the energy balance of the overall system. This article shows the basic concept how cubes are defined and how formal concepts for interfaces, system behaviour, and hierarchical layout are described.

Introduction

Balanced Manufacturing (BaMa, the project is running from 2014 until 2018) will develop a simulation-based tool for monitoring, predicting and optimizing energy and resource demands of manufacturing companies under consideration of the economic success factors time, costs and quality. Goal of the modelling approach - which is done in the first part of the project –should be the development of methods, which are able to integrate all building blocks of the facility (production, building, energy, logistics, management system) with one approach. This phase of BaMa started with a thorough system analysis and the definition of the methodology. In order to address these challenges, systematic approaches, as described by Thiede et al in "A systematic method for increasing the energy and resource efficiency in manufacturing companies" [1] have been analysed. A modular approach was chosen, that segments a production facility into so called "cubes". In the first step the features of the cubes were defined. Cubes have in addition clearly defined interfaces and represent a certain physical behaviour that contributes to the energy balance of the overall system. Nevertheless all cubes should be built up with the same architecture.

One of the main goals of BaMa is to monitor and compute energy and resources consumption. For doing so, based on the cube related energy and resource flow analysis, the method should be able to generate a specific product-footprint for every product running through the "cube system". The product footprint represents a products expenditures concerning cost, time, energy and the environmental impact such as resulting carbon emissions in the product life cycle phase within the factory.

Of course there are already comprehensive planning tools, such as [2], which also have been analysed. Regarding this analysis BaMa will also be implemented inside a customised toolchain. The toolchain (Balanced Manufacturing Control, BaMaC) allows energy efficient operation, design and refurbishment of production plants under competitive conditions, with regard to minimal energy and resource consumption. Tools to assist energy conscious steering of a plant during operation will be developed as suggested by K. Bunse et al in [3]. BaMaC will contain three core modules*:*

The modules in detail will be able to support the three tasks: *Monitoring:* data on resources consumption will be aggregated and visualised, data can be implemented into simulation of cubes.

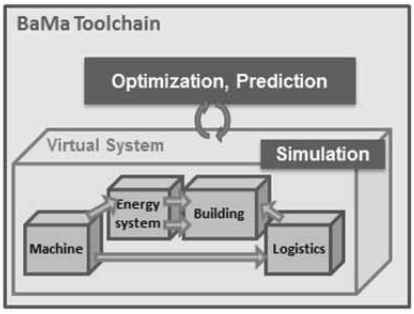

Figure 1. Future Modules of the BaMa Toolchain. The simulation approach has to fulfil various demands.

Prediction: allows forecasting of overall energy demand of the plant based on the product-footprint and the production schedule. *Optimisation:* based on data and numerical simulation-models of the cubes, this part of the tool chain will improve the plant operation with regard to the optimisation targets energy, time, costs and quality.

By integrating the four main optimisation fields building, energy system, production, and logistics equipment BaMa will be applicable to a variety of industrial sectors. It will serve as a basis for a software tool chain which will be integrated into industrial automation systems, such as ERP or MES. The toolchain will introduce energy efficiency as a steering parameter into the control centre, thus enabling manufacturing companies to balance energy efficiency and competitiveness in their continuous operation strategies.

To satisfy the described demands of BaMA and BaMaC the cube concept needs to fulfil a variety of characteristics. The concept has to fit a variety of applications i.e. it should be able to integrate all relevant building blocks of the facility (machines, energy system, logistics, …) with the same architecture. It is used as formal description of the real production plant and also as basis for models of the system. This modelling should be possible more or less “directly”, without much amount of work for translating. The cubes must have clearly defined features and interfaces and the system should be able to generate a specific product-footprint for every product running through the “cube system”. And finally of course implementation should be possible easy, fast and stable.

1 Motivation of BaMa - Footprinting

One of the most interesting demands – and main goal - in BaMa is the implementation of a comprehensive foot printing for industrial production plants. Industrial production accounts for 40% of the energy consumption of Europe, with an estimated potential for reduction of 30% to 65% [4]. A common top-down approach to identify the environmental impact of products is to assess the Carbon Footprint of Products (CFP) on a one-year-basis. This procedure is important for raising awareness. However, for the purpose of optimizing plant operation it is not well suited, because the results can vary on a large scale due to the lack of transparency of different methods [5], missing standardisation [6] and the lack of reliable data [7]. In addition the CFP fails to incorporate the diversity of different types of expenditure that go into the manufacturing of products.

In order to address these issues the BaMa bottom-up approach for aggregating a product footprint during the production phase of the product life cycle was proposed. This method allows for real-time evaluation of a batch or even single product using monitoring or simulation data. The definition of a significant footprint sets product success factors in context with its ecological impact. In particular energy, costs, carbon emission and time will be captured and visualised for the transformation process a product undergoes within the plant. Each part of the plant contributes to the product’s energy, cost or time consumption, as well as carbon emission, which accumulates the product footprint. The energy used by production machines, auxiliary infrastructure, logistics and the building is aggregated from the entry of the raw materials to the departure of the finished good. The integral footprint of all products produced in a year match the yearly carbon footprint of the plant exactly. So comparability with conventional studies is achieved.

From this bottom-up approach different challenges arise. For example, the incorporation of standby-, setup- and ramp-up times, the energy consumption of the administration and the allocation of different products and by-products manufactured at a machine are some of the problems. The necessity to calculate mean values and dividing them between different products demands for a way to assess the degree of which each product is responsible for the generated footprint. One can easily see that measurement of data for this applications and modelling of such processes is challenging. Implementation would strongly benefit of a clear defined modelling concept and approved, straight forward methods. The cube approach, in which the system is described through black boxes (cubes) connected through inputs and outputs has to manage to map the complexity of a manufacturing facility in the necessary detail and breaking down the plant into its elements.

The inputs and outputs of cubes can be material, energy or information flows. Energy flows carry a qualifier to determine the different expenditures, including carbon emission and monetary value. The products in the material flow accumulate the footprint by aggregating the cost, energy consumption, carbon emission and time inside the system boundary.

2 Requirements for Cubes

Based on the previous findings, a methodology for conducting a comprehensive system analysis of a production plant in preparation for the implementation of Balanced Manufacturing had to be developed. The methodology should be formulated at a generic level to ensure its usability in a variety of production facilities.

As described the basic element of this system analysis consists of the so called cubes. The idea was that cubes constitute subparts of a system "production-plant" and have the following properties:

- defined boundaries,
- interfaces to other cubes,
- a certain physical behaviour that contributes to the energy balance of the system
- and usually some degree of freedom to be influenced for optimisation.

To put it differently, the boundaries of sub systems in terms of energy-, material- and information flows had to be thoroughly defined to intersect the whole system into observable parts. The characteristics and attributes of cubes should be specified in a generic way in order to guarantee the applicability for all parts of the plant and for different kinds of productions. A cube could be a machine tool, a chiller, a baking oven, the production hall or a utility system. The definition of the cubes should allow implementing the described product-footprint evaluation, which sets the product success factors in context with its ecological footprint.

In particular the resources energy, costs and time will be captured and visualised for the transformation process a product undergoes within the plant. Each cube should contribute to the product's energy, cost or time consumption within the production plant which accumulates the product footprint. The product-footprint should be made up of a high number of originally independent data streams that are aggregated in a time-synchronised manner. So also methods for suitable data aggregation and fragmentation should be found and described.

So our approach leads us to the following process. (see Figure 2) . In the first step we analyse general systems of production plants. As a matter of fact in BaMa a number of basic applications of real world system were taken to be analysed (e.g. production facilities of semiconductors, bakeries, metal processing industries, …). Based on these approaches several specific cubes are defined with a variety of needed features for input, output, system behaviour, system variables, changing processes and many more. An additional general analysis is done and a generic cube definition is formulated. This cube definition is one step before the formalisation of the modelling concept we will introduce. The modelling concept (formal model) will especially need to be able to handle continuous and discrete processes running through the "cube system". The last step is the implementation of simulation applications for BaMaC.

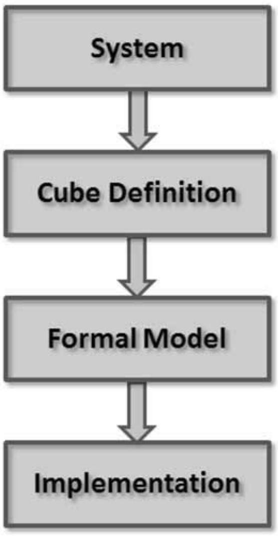

Figure 2. Analysis of the system (a variety of systems and their generalisation) leads us to the general "cube concept". This helps to formalise the real world and its control as well as future models. A formal model definition and implementation finalise the project phase.

Most important at this stage was the demand, that the cube concept should be as generic as possible not including specific model restrictions at that time. For these demands ontologies seem to fit in some kind of way. For this reason - and as a next step - the basic idea of ontologies, as well as the motivation for using such ontological analysis in modelling and the role in the modelling processes should be described.

3 Ontologies in Modelling

After all analysis of the requirements for cubes showed that ontological analysis could be a promising approach. The project team thought at that time, that probably the project will not need the whole range of possibilities, but some aspects seemed promising. Ontologies have been an effective tool in modelling and simulation to help to address some aspects in complex modelling & simulation projects.

To understand principles of the ontological approach and to estimate benefits and motivations for using Ontologies in modelling we relied on the work of Benjamin et al "Using Ontologies for Simulation Modeling" [8]. An ontology is an inventory of the kinds of entities that exist in a domain, their characteristic properties, and the relationships that can hold between them [9]. In our case the domain is the part of the actual world, which is a production plant. Such a production plant has its own ontology, which we refer as a domain ontology with some sub domains. In a domain ontology, we define various kinds of objects (e.g., machines and tools), properties (e.g., being made of metal), and relations between kinds and their instances (e.g., part of).

In general we need to extract the nature of concepts and relations in any domain and representing this knowledge in a structured manner. An ontology and its building differs from traditional modelling activities (adding information and data to a formal system description) not only in depth but also in breadth of the information used. As Benjamin et al describe in [8]: "Thus, an ontology development exercise will expand beyond asserting the mere existence of relations in a domain; the relations are "axiomatized" within an ontology (i.e., the behaviour of the relation is explicitly documented). Ontology development is motivated not so much by the search for knowledge for its own sake (as, ideally, in the natural and abstract sciences), but by the need to understand, design, engineer, and manage such systems effectively." For the cube concept, which should be used for various cube types within one model and as a basic library for future production plant models.

For defining ontologies different aspects are important as described in [10] especially determining the appropriate scope and granularity of ontologies and the use of ontologies as a basis for defining model repositories.

Inefficiency is often a problem in knowledge acquisition and management. Information that has been recorded before is captured again and modelling is done multiple times. Rather than having to identify information again and again in different applications, the idea of an ontology is to develop libraries " large revisable knowledge bases of structured, domain specific, ontological information in which can be put several uses for multiple application situations" [8].

The literature describes ontologies as important for modelling for a lot of reasons. Ontological analysis has been shown to be effective as a first step in the construction of robust knowledge based systems [11]. Modelling and simulation applications can take advantage of such technologies. As a second point, ontologies help to develop standard, reusable application and domain reference models. This characteristic seemed to fit for integration of various production plant types. Last but not least ontologies are at the heart of software systems that facilitate knowledge sharing.

Motivation for Using Ontologies in Modelling

Basic motivations for using ontologies in modelling and simulation are that they are useful across the modelling and simulation lifecycle, particularly in the problem analysis and conceptual model design phases. They play a critical role in simulation integration and simulation composability and they are important in facilitating simulation model interoperability, composition and information exchange.

One of the key ideas is to allow the decomposition of the overall system model into smaller, more manageable components, and to distribute the model development effort among different organisations or functional groups [12]. This is a perfect approach for the planned cube concept. Once the component simulation models have been developed, there is a need for mechanisms to assemble a simulation model of the entire target system in a manner that the "whole (system) = sum of its components."

An important challenge is modelling and simulation composability (from a set of independently developed components). "Composability is the capability to select and assemble simulation components in various combinations into simulation systems to satisfy specific user requirements" [13]. Composability enables users to combine, recombine, and configure or reconfigure components in numerous ways to satisfy their diverse needs and requirements.

There are two forms of composability: syntactic and semantic . Syntactic composability deals with the compatibility of implementation details such as parameter passing mechanisms, external data accesses, and timing mechanisms. Semantic composability, on the other hand, deals with the validity and usefulness of composed simulation models [13].

As a matter of fact these advantages of ontological analysis seemed to perfectly fit the needs of our cube concept and the formal modelling process afterwards. The process described in Figure 2 was perfectly set for application of the basic ideas of ontological analysis.

Role of Ontologies in the Modelling Process

Simulation models are often designed to address a set of modelling objectives or to answer a set of questions. An important first step in simulation modelling is to define the purpose of the model. This activity involves several related activities. On one hand the developer gets a "list" of not formalised problem symptoms. The domain experts often describe a problem in terms of a list of observed symptoms or areas of concern. The desire is to identify the cause of these symptoms and to suggest remedies. As described in chapter 1 one of the main objectives for the cube approach is to introduce the possibility of bottom up foot printing for production plants and to identify the origin of those symptoms. In addition often the domain experts specify the objectives of a project in terms of a specific question that needs to be answered, or, alternatively, specifies explicit goals to be met. For instance, in our example the manager of the production plant might ask the question "How can I optimise my production process?" or state a goal: e.g., "I need to reduce used energy by 20% on all my machines.". Using clearly defined objectives can help a lot in both cases to formalise and structure the described goals.

The purpose of the model also depends on constraints on possible solutions to the problem. The domain expert, based on past experience with similar situations, often suggests a variety of possible alternative solutions that must be explored. For example, a production plant manager who would like to increase production rate may, because of a budgetary constraint, be unwilling to invest in new machines, but may instead be able to hire additional labour. Ontologies will help facilitate the above tasks as well.

The advantages and also the justification of investing additional resources needed for following an ontological approach instead of doing only the work which is unconditional are on one hand providing a mechanism to interpret and understand the problem descriptions. Domain experts often use specialised terminology to describe symptoms and problems. Domain ontologies help with the unambiguous interpretation of the problem statements and in precisely conveying information about the problem to the simulation modeller. Cube can – in a reduced way – fulfil these characteristics. In addition harmonizing statements of objects that are described from multiple perspectives (often, this is a non-trivial task because of terminological differences and the lack of explicit descriptions of the semantics of different terms and concepts – see also [8]). Last but not least the ontological analysis unambiguously interprets limiting constraints that need to be addressed relative to accomplishing project goals.

All together the BaMa Cube concept will not fulfil all formal needs and demands of an ontology. As a matter of fact within BaMa the ontological approach was identified to support various needs of the modelling process. It helps in the process of getting "axiomatized" rules for the modelling of production plant sub systems. So the behaviour of the relations between subsystems is explicitly documented as well as the possibility how and what to "footprint". Objects, properties and relations are clearly defined and are reproducible for every simulation project, that will be implemented with the cube concept. BaMa will not only generate "one model of one production plant" but will develop libraries and large revisable knowledge bases of structured, domain specific, ontological information in which can be put several uses for multiple application situations. In practice scope and granularity of the cubes can be defined clearly and can also be supervised. By using ontological analysis decomposition of the overall system model into smaller, more manageable components is done as well as distribution of the model development will be possible. The aim of composability enables future users of BaMa to combine, recombine, and configure or reconfigure components in numerous ways.

4 Cube Definition

On basis of the above described ideas the generic term "cube" describes an encapsulated part of the observed overall system (domain). This is part of a methodological approach to address the high system complexity and heterogeneity by dividing the overall system from an energetic point of view into well-defined manageable modules (see Figure 3), which then allow a focused system analysis independent form the surrounding environment. Integrating different viewpoints and areas of engineering (machinery, energy system, building, and logistics) in a single system description can be interpreted as combining a number of ontological sub-domains and makes it necessary to establish a general specification of the cube properties and interfaces.

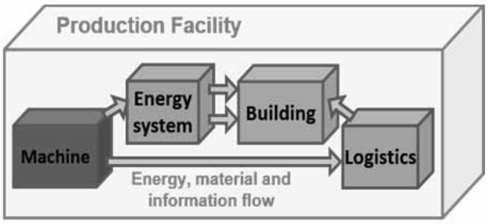

Figure 3. Production facility as interacting cubes.

The cubes consolidate all information and resource flows (energy, materials, etc.) within identical system boundaries, which not only promotes transparency during simultaneous analysis of energy and material flows, but the obtained modularity also increases flexibility for adaptation to specific environmental conditions.

Cubes have uniformly and consistently defined interfaces through which they interact with each other by exchanging energy, material and information flow, see Figure 2.

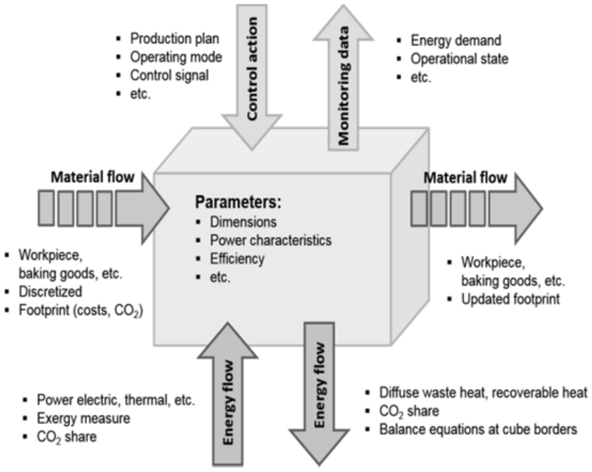

Figure 4. Generic cube interfaces with energy, material and information flows.

The material flow incorporates the immediate value stream (e.g. work piece, baking goods) and is described as discrete entities.

All necessary energy flow (electrical, thermal, etc.) is represented as continuous variables together with their respective CO2 rates and is quantified inside the cube boundaries using balance equations. Information flow provides operating states and monitoring values for the higher-level control as well as control actions for the cube module

This modular cube description and specified interfaces then enables analysing and modelling the internal behaviour independent from its surroundings. For experimental analysis based on measurement data, cube interfaces can be equipped with measuring devices to detect incoming and outgoing flows. Also, experimental production cubes are being constructed which allow a more in-depth energy analysis and the inclusion of more detailed measurement information for developing data models and usage in simulation.

The modularisation of the observed overall system is not only used for developing simulation models for these systems. So the cubes have not only the "virtual simulation block" (so-called virtual cube, see Figure 5) in the form of a component in a simulation model, which we have to formalise later on but also the representation in the "real world" e.g. in the automation system of the production plant.

The retained encapsulation and interaction via defined interfaces provides flexibility during internal modelling of the cubes (e.g. as mathematical models, data models, etc.) and for reusing implemented components in other models.

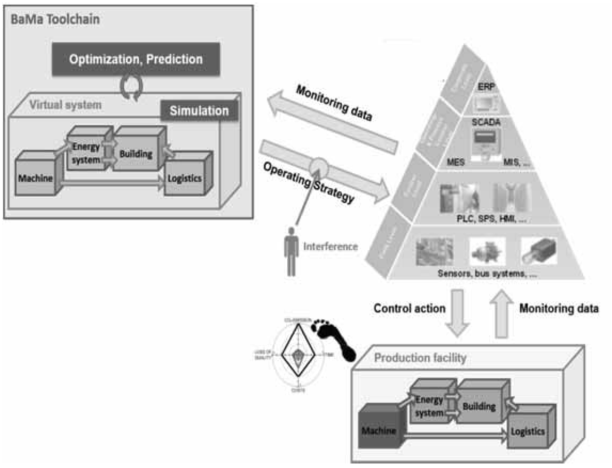

Figure 5. Architecture of the BaMa toolchain including the production facility in the "real world" and the a virtual representation of the observed system (simulation).

Figure 5 shows the relationship between real and virtual cubes in the simulation environment and the integration into the overall automation system architecture. The BaMa toolchain obtains measurement and status data from different levels of the automation system and on the other hand delivers prediction data and proposals for optimised operation strategies that can be adopted - with user interaction - in the real system.
The generic interface and attributes definition of the cubes serves as a basis for specifying four cube categories (see Figure 6).

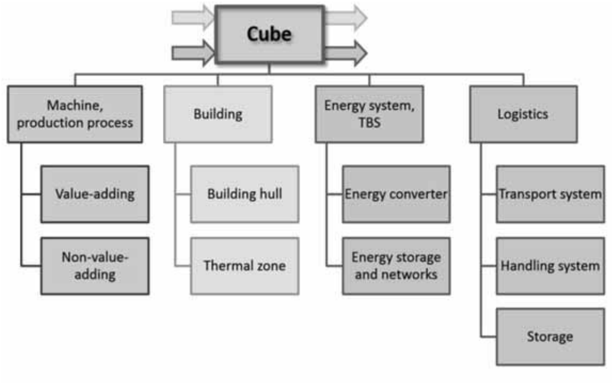

Figure 6. Categories and subcategories of cubes.

Defining the cubes succeeded in the possibility to have reusable modules for representing machines and all other physical inventory within a production plant. Both discrete and continuous flows can pass through the system. The modules are on basis of one methodology (all cubes are children from a master cube, see Figure 6) and can so be implemented in the same way. A more detailed description of the cube methodology and the individual cube categories can be found on the cube subsection on the BaMa project website with the address (http://bama.ift.tuwien.ac.at/). As a matter of fact while doing the cube concept, the modelling group of BaMa always had in mind how to formalise in the next step the model libraries on basis of the given features and interfaces, which was helpful in the next step.

5 Formalisation of Cubes

After the generic description of cubes the question of implementation arises. As far as described we combined various areas of production plants, where entities are able to pass from one area to the other. Still we need to be able to generate the planned foot printing.

As described in the last chapter on the one hand, the modelling approach needs to provide solutions for hybrid systems, i.e. systems containing continuous as well as discrete parts. Of course there are many software tools which offer solutions for either continuous or discrete models but not for combined models. Still, there are a few commonly known simulation environments like Simulink or Modelica who allow the combination of discrete and continuous model parts. In the case of Simulink, for example, discrete SimEvents models can be combined with continuous models described by ordinary differential equations (ODEs) where the SimEvents scheduler and the ODE solver work in parallel and cooperate, which seems to work fine for very simple trials, but as soon as large or rather complex systems are implemented, the simulation can fail due to non-resolvable errors. Additionally, the execution of actions intended to take place at the same time an event occurs has to be defined by the user right before or right after the event in order to prevent unintentional results.

On the other hand in the BAMA project buildings as well as machines, building services and logistics have to be modelled and simulated on the whole in spite of their different requirements regarding modelling approaches and simulation techniques. As this is virtually impossible to realise in one tool alone, the most common way to face this task is to use cooperative simulation (co-simulation). There exist some co-simulation tools developed especially for systems containing buildings and machines, but most of them regard mainly thermal processes and perhaps energy consumption but disregard resources and do not support optimisation. Furthermore these tools in general gravely restrict the software used for partial models.

These problems were approached by taking the step between the generic description (Cube Definition) and the actual Implementation - using a simulation formalism (Formal Model) – see Figure 2. In 1976 Bernard Zeigler proposed in his book “Theory of Modeling and Simulation” [14] a classification of dynamic system-models into three basic types: Discrete Event -, Discrete Time – and Differential Equation – systems (DEV, DTS, DES). DEV are usually simulated using an event-scheduler, DTS are system models where changes of state-values are happening in equidistant instances of time and DES as purely continuous models, described with differential equations. Zeigler introduced system-specification-formalism for all three types (DEVS, DTSS and DESS) where DTSS is a subtype of DEVS.

Very important properties of the formalisms are their hierarchical nature and their closure under coupling which perfectly fits the cube features. That is, an atomic model of each formalism has inputs and outputs, which can be coupled with inputs and outputs of other atomic blocks or with the inputs and outputs of an overlying non-atomic model which inhabits these atomic models (hierarchical). The resulting overlying model now behaves exactly like an atomic model (closure under coupling) of the particular formalism and therefore again can be coupled with other atomic and non-atomic models. In the following part we assume the knowledge of atomic and coupled DEVs and atomic and coupled DESs (see [14].

On basis of these atomic and coupled DEVS and DESS Zeigler introduced an additional formalism called DEV&DESS [15] standing for Discrete Event and Differential Equation System Specification. DEV&DESS is intended to describe so called hybrid system. In this context, hybrid system means a system consisting of both, a discrete and a continuous part, which is exactly what is needed for cubes. Atomic DEV&DESS systems can be described with the system $DEV\&DESS_{atomic} = < X^{discr}, X^{cont}, Y^{discr}, Y^{cont}, S, \delta_{ext}, C_{int}, \delta_{int}, \lambda^{discr}, f, \lambda^{cont} >$ where $X^{discr}, Y^{discr}, X^{cont}, Y^{cont}$ describes a set of possible discrete and continuous inputs and outputs and $S = S^{discr} \times S^{cont}$ is a set of possible states, which describes the state space. Together with $Q = \{(s^{discr}, s^{cont}, e) \big| s^{discr} \in S^{discr}, s^{cont} \in S^{cont}, e \in \mathbb{R}_0^+\}$ we get $\delta_{ext}: Q \times X^{cont} \times X^{discr} \rightarrow S$ and $\delta_{int}: Q \times X^{cont} \rightarrow S$ as internal and external state transition function, $\lambda^{discr}: Q \times X^{discr} \rightarrow Y^{discr}$ and $\lambda^{cont}: Q \times X^{cont} \rightarrow Y^{cont}$ as discrete and continuous output function as well as $f: Q \times X^{cont} \rightarrow S^{cont}$ as rate of change function ("right side" of an "ODE-System") and $C_{int}: Q \times X^{cont} \rightarrow \{true, false\}$ as state event condition function.

As described above the DESS and DEVS formalisms are well known in literature. In our case we focus on the additional meaning of C_{int}. C_{int} is a function of the actual state q and continuous input value $x^{cont}(t)$ and is responsible for triggering internal events, which then may cause a discrete output $y^{discr} = \lambda^{discr}(q, x^{cont})$ and definitely results in the execution of δ_{int}. Therefore, internal events in DEV&DESS are not exclusively dependable on time, as it is the case with DEVS, but may also be triggered because of the system state S reaching a certain threshold. Events of the later type are called state-events.

Since the state transition functions δ_{int} and δ_{ext} update the whole state, including its continuous part, they may lead to a discontinuous change in s^{cont}. Thus, as s^{cont} is the output of an integrator, this integrator needs to be reseted, each time an external or internal event occurs.

The last distinguishing feature of the whole, DEV&DESS, to its components DEVS and DESS is the dependency of δ_{int} and λ^{discr} of the actual continuous input value. For DEV&DESS to be well defined, we need to fulfil both, the requirements for the DEVS part, and the requirements for the DESS part. Therefore for each possible input-trajectories and initial states, during a finite time interval only a finite number of events is allowed to happen, the function f again has to meet the Lipschitz requirements and the continuous input and output signals need to be bounded and piecewise continuous.

Coupled DEV & DESS $N =< X^{discr} \times X^{cont}, Y^{discr} \times Y^{cont}, D, \{M_d\}_{d \in D}, \{I_d\}_{d \in D \cup \{N\}}, \{Z_d\}_{d \in D \cup \{N\}}, Select >$ are described via $X^{discr}, Y^{discr}, X^{cont}, Y^{cont}$ as a set of possible discrete and continuous inputs and outputs, D as a set of involved "child-DEV&DESS"-denominators, M_d as child DEV&DESS of N for each $d \in D$, $I_d \subset D \cup \{N\}$ influencer set of $d, d \notin I_d$ and finally together with Z_d as interface map for d and $Select: 2^{D \cup \{N\}} \rightarrow D \cup \{N\}$ as tie-breaking function we get the whole system.

The meaning of all the terms listed above are already known, either from the atomic DEVS definition or from coupled DEVS or coupled DESS systems. But there are some restrictions, concerning the coupling of discrete outputs with continuous inputs and vice versa. At first, we divide the interface map Z_d into two component functions. One for the calculation of the discrete inputs of block d $Z_d^{discr}: YX_i \rightarrow XY_d^{discr}$ and one for the calculation of the continuous inputs $Z_d^{cont}: YX_i \rightarrow XY_d^{cont}$. for each $i \in I_d$

Second, we need to define, how to interpret a connection from an discrete output to an continuous input and the other way round: Discrete output signals, actually are only existent at instance of time, where they are produced. The rest of the time, the value of the output-signal is the empty set ∅ or non existent. However, to enable connections between discrete outputs and continuous inputs, we define discrete outputs to be piecewise constant. So the value of a discrete output at a time between two output-events is always the value of the last output-event. Therefore it is allowed to connect discrete outputs arbitrary to continuous inputs. The other way round isn't that easy, and it is necessary to apply restrictions. Thus, continuous outputs are only allowed to be connected to discrete inputs, if they are piecewise constant. One could think of a connection from discrete to continuous being realised by putting an additional DEV&DESS-block in between, that receives the discrete output at its discrete input and forwards it to its continuous output. The other way around works too.

As DEV&DESS sums up the functionality of both sides, the discrete and the continuous one, the modeller has to deal with the requirements of each formalism as well. On the one hand, the modeller needs to take care, not to produce algebraic loops and on the other hand he also needs to think of how to define the tie-breaking function select for the model to produce the desired behaviour. As Zeigler showed [15], all three basic formalism, DEVS, DTSS(already included in DEVS) and DESS describe subclasses of the set of DEV&DESS-describable systems. Therefore DEV&DESS-describable is perfectly suited to formally describe and simulator-independently hybrid models of real systems. In our case - as a step in between - we used the cube formalism as organisational structuring of the modelling process using ontological analysis know how. Every cube has continuous inputs like various forms of energy, which are part of a continuous model, and many cubes, like machine cubes handling work pieces, have discrete inputs which are handled in a discrete system part of the machine model.

Since the DEV&DESS formalism does not specify solution methods, solution algorithms for the discrete part and differential equation solvers for the continuous part can be chosen at the point of implementation. In the case of cubes comprising purely continuous models, the DESS formalism can be applied and still linked with other cubes described by DEV&DESS or DEVS for plain discrete systems. Additionally, several atomic DEV&DESS can be embraced by another DEV&DESS called coupled DEV&DESS afterward for even better structuring; hence the DEV&DESS formalism also fulfils the hierarchy requirement, which represents an obligatory demand in the BAMA cube definition.

As every DEV&DESS, be it coupled or atomic, can be regarded as separate systems and each DEV&DESS represents one cube in which the balance equations consider everything within the cube's borders, which are per definition balance borders, closure regarding balance equations can also be ensured as long as the generic description of the cube can guarantee it.

DEVS is a very general formalism. As a result, it can be shown, that a lot of other discrete-event-formalism, as for example Event-Graphs, State charts, Petri-Nets and even Cellular Automata describe subclasses of the set of all systems describable by DEVS. That's why Zeigler proposes the so called DEVS-Bus as common interface for multi-formalism simulation. For implementation and formalisation this keeps the possibility of a "general approach" for integrating domain experts knowledge in future approaches and involve possible additional model concepts (e.g. additional cubes shall be described in one of the ways mentioned above).

6 Implementation

Last but not least, since digital computers only are able to work in a discrete way, discretisation is necessary for each DEVS and DESS-part of a DEV&DESS to be able to be simulated on a digital computer. For pure DESS-models, usually ODE-solver-algorithms are used, to numerically solve the differential equations, i.e. to simulate the DESS model. Therefore, the DESS model in combination with the used ODE-solver constitutes a DEVS model, approximating the DESS model. This resulting DEVS model, as each DEVS model, can then be simulated error-free on a digital computer, apart from the error due to the finite representation of real numbers.

But due to the fact that the DEV&DESS formalism is, as its name implies, just a formalism, it is independent from the implementation software. This is very important for the BAMA project since a lot of participating industry partners already use certain automation software which is intended to be able to communicate with the simulation software and every developing partner has preferred simulation tools or limited licenses.

The DEV&DESS formalism does not restrict the possibilities for the cube interfaces. In the cube definition described briefly above it has been defined that input and output signals can be arrays and may represent physical values which carry a unit or other attributes ensuring consistency. This is possible with the DEV&DESS formalism since the only specification for inputs or outputs to a DEV&DESS is that there is a set of discrete and/or a set of continuous inputs and outputs. Hence the demands on cube interfaces can be met by the DEV&DESS formalism. Finally taking a deeper look at ontological analysis was worth doing, even if BaMa did not implement its own ontology. Defining and implementing the process as described below (see Figure 7) was one of the keys to successfully implement the cube methodology in the first phase of BaMa.

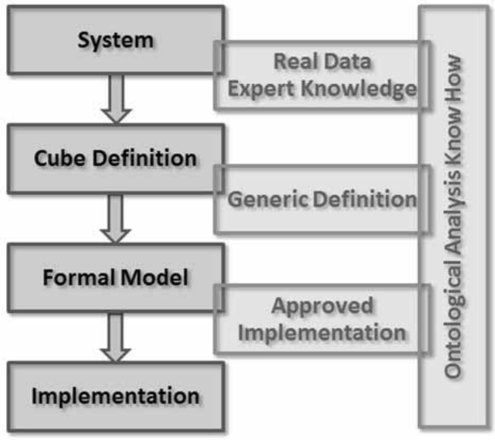

Figure 7. System Analysis and Modelling uses Ontological Analysis Knowhow for reusable, quality assured results.

At the actual point of BaMa the definition of the DEV&DESS formalism is finalised. As a matter of fact there is still a link missing to get to the implementation itself, but on the one hand there exist several tools implementing the DEV&DESS formalism with a certain approach like PowerDEVS using QSS for the discretisation of the DESS parts and thus transforming DEV&DESS into DEVS only, QSS-Solver with the Micro-Modelica language, M/CD++, or a Simulink library for DEV&DESS developed at the Hochschule Wismar or DEVS-only tools like DEVS-Suite, CD++ and JDEVS; on the other hand in the course of the BAMA project several typical scenarios have already been formalised with the DEV&DESS formalism and implemented PowerDEVS for test purposes, so it is warranted that this formalism can actually be used as a bridge from the BAMA cube definition to the BAMA implementation.

References

[1] S. Thiede, G. Bogdanski, C. Herrmann, “A systematic method for increasing the energy and resource efficiency in manufacturing companies” Procedia CIRP, Vol. 2, pp 28 – 33, 2012

[2] F. Bleicher, F. Dür, I. Leobner, I. Kovacic, B. Heinzl, W. Kastner, „Co-simulation environment for optimizing energy efficiency in production systems“ CIRP Annals – Manufacturing Technology, Vol. 63, pp 441 – 444, 2014

[3] K. Bunse, M. Vodicka, P. Schönsleben, M. Brülhart, F. Ernst, „Integrating energy efficiency performance in production management - gap analysis between industrial needs and scientific literature” Journal of Cleaner Production, Vol. 19, pp. 667 – 679, 2011

[4] E. Bonneville, A. Rialhe, “Good practice for energy efficiency in industry”, http://www.leonardo-energy.org/sites/leonardo-energy/files/root/Documents/2009/DSM-industry.pdf (last accessed Dezember 15, 2014).

[5] J. P. Padgett, A. C. Steinemann, J. H. Clarke, M. P. Vandenbergh, “A comparison of carbon calculators”, Environmental Impact Assessment Review 28, 2008, pp 106–115.

[6] M. Gaussin, G.Hub, S.Abolghasem, S.Basu, M.R. Shankar, B.Bidanda, “Assessing the environmental footprint of manufactured products: A survey of current literature”, Int.J.Production Economics 146, 2013,pp. 515–523.

[7] R. Neugebauer, E. Westkämper, F. Klocke, A. Kuhn, M. Schenk, A. Michaelis, D. Spath, E. Weidner, “Energieeffizienz in der Produktion - Untersuchung zu Handlungs- und Forschungsbedarf“, FhG, 2008, pp. 350-351.

[8] P. Benjamin, M. Patki, R. Mayer, “Using Ontologies for Simulation Modeling”. Proceedings of the 2006 Winter Simulation Conference, L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds., pp. 1151-1159, 2006

[9] P. Benjamin, C. Menzel, and R. J. Mayer.Towards a method for acquiring CIM ontologies. International Journal of Computer Integrated Manufacturing 1995, 8 (3), 225–234.

[10] P. Fishwick and J. Miller. Ontologies for modeling and simulation: Issues and approaches. Proceedings of 2004 Winter Simulation Conference. Piscataway, New Jersey: Institute for Electrical and Electronics Engineers.

[11] J. Hobbs, W. Croft, T. Davies, D. Edwards, and K. Laws. The TACITUS Commonsense Knowledge Base, Artificial Intelligence Research Center, SRI International, 1987

[12] P. Benjamin and M. Graul. A framework for adaptive modeling and ontology-driven simulation. Proceedings of SPIE, Enabling Technologies for Simulation Science X, Vol.6227, 2006

[13] M.D. Petty and E. W. Weisel. A composability lexicon. Proceedings of the Spring 2003 Simulation Interoperability Workshop , Orlando FL, March 30- April 4 2003, pp. 181-187.

[14] B. Zeigler B. "Theory of Modeling and Simulation", Wiley-Interscience Publication, John Wiley & Sons, 1976

[15] B. Zeigler, H. Praehofer and T.G. Kim, “Theory of Modelling and Simulation”, Second Edition:, Elsevier Academic Press, 2000.

Acknowledgement. The project BAMA (Balanced Manufacturing) is funded by the Austrian Research Promotion Agency (FFG, project number 840746) and the Austrian Klima- und Energiefonds (KLIEN).