INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED15
27-30 JULY 2015, POLITECNICO DI MILANO, ITALY

MODELING FACTORY SYSTEMS USING GRAPHS - ONTOLOGY-BASED DESIGN OF A DOMAIN SPECIFIC MODELING APPROACH

Plehn, Christian; Stein, Florian; Reinhart, Gunther
Technische Universität München, Germany

Abstract

Changeable factory systems are a viable strategy for manufacturing companies to cope with dynamic and uncertain environments, characterized by frequent engineering changes, product and technology innovations, and continuous improvement initiatives often resulting in changes of factory systems. Flexibility and changeability are considered beneficial properties helping to be prepared for the various possibilities of an uncertain future. To support the analysis of these system properties, suitable modelling techniques are required covering both structural and element properties. Hence, the objective of this paper is to provide a graph-based domain specific modelling approach for factory systems. Metamodels for nodes and edges are suggested based on metamodel and ontology design theory and an extensive review of factory planning literature. The approach is demonstrated by modelling a simple compressor shaft workshop production. Finally, promising application perspectives of the graph-based modelling approach are outlined.

Keywords: Metamodel, Graph, System properties, Systems engineering (SE), Manufacturing

Contact:
Christian Plehn
Technische Universität München
Institute for Machine Tools and Industrial Management
Germany
christian.plehn@iwb.tum.de

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

1.1. Changeable systems in a dynamic world

Designing socio-technical systems has become an increasingly challenging task. Researchers in the domains of design as well as manufacturing have spent tremendous effort on understanding the drivers of system changes. Investigations in effective and efficient management of change processes and the design of easily changeable products or manufacturing systems are of further interest. With increasing effort, since the early 1970s manufacturing science and operations management have been approaching the concept of changeability particularly by looking through the lens of manufacturing flexibility (de Toni and Tonchia, 1998). Research has been conducted to define suitable frameworks to analyse different facets of flexibility and to find ways for their accurate measurement. Besides, a variety of empirical work has investigated correlations of manufacturing flexibility with organisational attributes, technology, and measures of firm performance (Vokurka and O'Leary-Kelly, 2000). For about fifteen years, more recent work in manufacturing science shifted its focus from static flexibility to dynamic flexibility which is often referred to as changeability or transformability, describing a generic potential of a manufacturing system to cope with unpredictable future challenges, e.g. fluctuating sales volume, new products, upcoming production technologies, and changing customer requirements. Different so-called "enablers" of changeability have been identified, such as modularity, scalability, mobility, compatibility, and universality of factory systems and their elements (Wiendahl and Hernández, 2006; Koren et al., 1999). When comparing the reasoning for the importance of changeability in engineering and manufacturing systems as well as the principles for hedging against future uncertainties, similar challenges and concepts can be identified. Representing the perspectives of product development, Fricke and Schulz (2005) name dynamic marketplaces, technological evolution, and variety of environments as major drivers for systems development. To define changeability of system architectures, they suggest the four aspects robustness, flexibility, agility, and adaptability which are again supported by basic (e.g. ideality, modularity) and extending principles (e.g. integrability, scalability, and decentralisation) (Fricke et al., 2000; Fricke and Schulz, 2005). In manufacturing literature the corresponding situation is commonly referred to as a turbulent manufacturing environment, characterised by both increasing complexity and dynamics originating from inside (e.g. markets, politics, capital market) and outside (e.g. technologies, products, human resources) a company (Wiendahl et al., 2007). Finally, it must be noted that changeability always results from an interplay of more specific interrelated principles (cf. e.g. Fricke and Schulz, 2005; Ross et al., 2008). These strategic system properties often share a uniform suffix, therefore they are commonly referred to as "ilities" (Ross, 2008; de Weck et al., 2011).

1.2. Factory system definition

In manufacturing literature, the layer model of production systems has been established (see e.g. Wiendahl et al., 2007). Figure 1 shows the resource view of this layer model distinguishing network, factory, segment, line, station, and technology level. However, for the analysis of sub-sections of a manufacturing plant this separation of levels appears too granular. Another issue arises from the use of the system term as a layer of the model itself while this concept should be thought independent from the level of abstraction. Hence, the following definition for factory systems is proposed: Factory systems comprise the spatial arrangement, relations, and properties of technology, personnel, and infrastructure in a differentiable sub-section of a manufacturing plant. The system boundary can be drawn depending on technological or product-oriented deliberations.

1.3. Model-based analysis of factory system "ilities"

Partly driven by the increasing industrial relevance of product service systems, both engineering and manufacturing systems are more and more understood as complex socio-technical systems, calling for a trans-disciplinary exchange of theories, methods and results. Especially strategic properties of these systems, the system ilities, can be interpreted and analysed for a wide range of system types. Major contributions towards a definition and theory of ilities as well as an investigation of their relationships and utility for protection against uncertainty and changing environments have been made by McManus et al. (2007), Ross (2008), Ross et al. (2008), de Weck et al. (2012), and Chalupnik et al. (2013). With the existence of unambiguous definitions of ilities, they have become accessible for analysis and evaluation. First attempts to make use of methods from structural complexity management (cf. Lindemann et al., 2009) to interpret selected ilities by means of structural criteria have been made by Maurer et al. (2014). Nevertheless, as ilities of systems arise from both elements and their interaction (Chalupnik et al., 2013), a complete interpretation can only be achieved including element attributes as well as structural metrics. However, suitable modelling techniques that capture relevant information of manufacturing systems and enable quantitative analysis of strategic system properties are still missing.

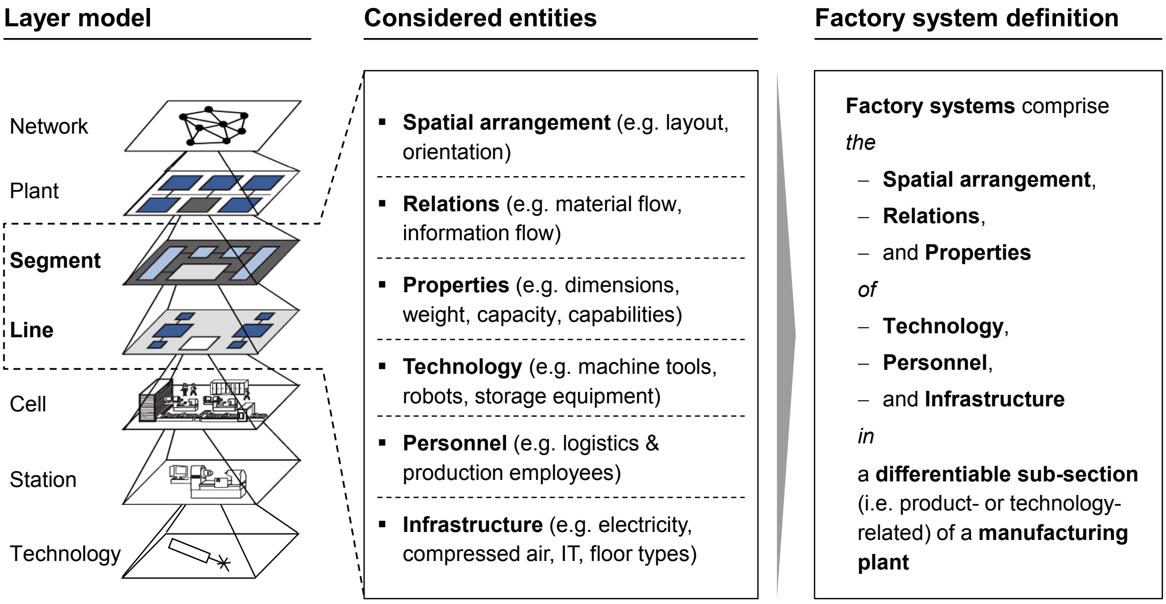

Figure 1. Layer model of production systems and factory system definition

1.4. Objectives and research methodology

The objective of this research is to provide a modelling approach designed to enable analysis of ilities for socio-technical systems within manufacturing plants - so called factory systems. In order to do so, a domain specific graph modelling language is suggested based on metamodel and ontology design theory. The domain specific language is represented by metamodels for nodes and edges. Research activities have been guided by the following research questions:

Which level of abstraction of factory systems is suitable for the aspired analysis of ilities?

Which requirements have to be fulfilled by the modelling approach to enable the analysis of ilities?

Which elements, relations, and attributes should be included in a metamodel and how should its class structure be designed?

The Design Research Methodology (DRM) documented by Blessing and Chakrabarti (2009) provided guidelines for the work carried out. According to the DRM framework, the research design at hand can be classified as a 'Type 3' where 'Research Clarification' and 'Descriptive Study I' are conducted 'review-based' while 'Prescriptive Study' and 'Descriptive Study II' are performed 'comprehensively'. In addition to a literature review, a comprehensive study requires "a study in which the results are produced by the researcher, i.e., the researcher undertakes an empirical study, develops support, or evaluates support" (Blessing and Chakrabarti, 2009). To support a subsequent evaluation of results, the success criterion of this work is defined as 'successful development of a design support for modelling factory systems suitable for the analysis of system ilities'.

The remainder of this paper starts with a short review of different fields of application for modelling in manufacturing science before requirements for factory system modelling are formulated. Following, a short comparison of existing languages and methods for analysis-oriented modelling of factory systems is made to motivate the usage of the domain specific approach chosen in this paper. After a short theoretical introduction to metamodel and ontology design theory two metamodels are suggested and finally used for modelling an exemplary factory system producing a compressor shaft. The paper concludes with a discussion of potential applications and outlining opportunities for further research.

2 MODELLING FACTORY SYSTEMS

2.1. Requirements of factory system modelling

A variety of modelling approaches exist devoted to the modelling of manufacturing plants on different levels of detail in order to to provide support for planning, analysing, visualising, or optimising the respective object under consideration. Some of the most prominent use cases in operations management are optimisation of layout (e.g. spatial arrangement of machines), material flow (e.g. capacity restrictions), production planning and control (e.g. scheduling), business processes, and energy efficiency. However, models to evaluate strategic system properties are rarely to be found in manufacturing literature, albeit research effort has been spent on quantifying individual ilities (e.g. modularity). This work is an attempt to enable future research in this field by providing a suitable modelling approach for socio-technical factory systems.

Requirements have to be specified in order to characterize the scope of modelling and to ensure that the resulting design support is constructed according to its intended purpose. Because "ilities are systemic properties that arise not only from the parts of a system, but also from the interactions between them" (Chalupnik et al., 2013), models need to enable structural analysis of the whole system as well as investigations of elements and properties. Since it is very likely that the metamodels will have to be expanded by later users, they should be designed adaptable. Furthermore, the modelling approach needs to enable an automated (i.e. machine-readable) quantitative analysis because of the complexity of real-world factory systems. Finally, keeping the industrial application in mind, it is of utmost importance to provide an attractive and intuitively understandable visual representation of the models as well as to guarantee an acceptable effort for their generation.

2.2. A short overview of methods and tools for system modelling

A common software or system design process often requires hundreds of participants with different backgrounds, leading to a tremendous complexity regarding information flow, relationships and other interdependent variables (Steward, 1981; Wang et al., 2014). In the past, different methods and tools to support this process have been developed. In the following, SysML, Object-Process Methodology (OPM), Design Structure Matrix (DSM) and graph-based domain specific modelling (GDSM) will be outlined briefly as these "languages" have proven their benefit for model-based systems engineering (de Weck et al., 2011).

SysML was developed on the basis of UML and is a standardized, graphical modelling language that is able to represent requirements, behaviour, structure and properties of systems and their components. With the mentioned abilities SysML supports the specification, design, analysis, verification as well as validation of complex systems (Valilai and Houshmand, 2009; Debbabi et al., 2010).

Like SysML, OPM is also derived from UML but uses a reduced set of building blocks and only one unified type of diagram cutting the effort for generating, synchronizing and maintaining a plenitude of diagrams for system and function modelling (Dori, 2002; de Weck et al., 2011).

Another technique to design, manage and analyse - particularly the structure - of complex engineering systems is the DSM. The DSM "is a square N x N matrix, mapping the interactions among the set of N system elements" highlighting a system's architecture (Eppinger and Browning, 2012). Due to the straightforwardness and flexibility of its concept it is applied in a variety of domains. Among others, DSM are also used in project management in order to improve planning, execution and management of complex projects by focusing on the optimization of information flows (Steward, 1981; Gunawan, 2012). Furthermore, DSM are widely used in the field of software development (Wang et al., 2014). For a comprehensive presentation of DSM applications see Eppinger and Browning (2012).

Network (referred to as graph in the following) and matrix approaches are dual formulations of a system's structure (de Weck et al., 2011). Graphs consist of nodes and edges which can be directed or undirected where nodes represent entities while edges are used to model any kind of interrelations. In general, nodes of a graph are treated equally, resulting in a highly abstracted representation of a system which is considered as a major drawback for the application in engineering. However, when it comes to visualization, statistical analysis, architectural properties (i.e. graph metrics), and big data graph-approaches demonstrate their benefits (cf. de Weck et al., 2011).

GDSM try to capitalize on the advantages of graphs while reducing the level of abstraction because a class structure of nodes and edges is allowed. These classes are designed specific to a certain domain.

As a result, GDSM stand in contrast to a general-purpose languages like UML or SysML (France and Rumpe, 2005).

2.3. Interim conclusion: Benefits of domain specific modelling approaches

As stated above, the application of a GDSM has several advantages. Beside those mentioned previously, the GDSM is customized for a certain problem (Giachetti et al., 2009), thus its information content can be tailored according to the intended "resolution" and aspects of system analysis. Also, communication among users within a domain is simplified. Finally, it has a restricted semantic scope reducing learning efforts and leading to increased usability. On the other hand, drawbacks are the limitation to a specific domain and the non-existence of standards (France and Rumpe, 2005). As a consequence, the decision about using a GDSM must be based on weighting up learning effort, modelling effort, and requirements of system analysis. Taking the named advantages into account, the development of a GDSM is pursued here.

3 METAMODEL DESIGN FOR FACTORY SYSTEMS

3.1. Metamodelling

3.1.1. Fundamentals

According to Paige et al. (2014) "a model is a formal description of phenomena of interest, constructed for a specific purpose, and amenable to manipulation by automated tools." In other words, models are tools to describe the structure, behaviour and other properties abstracting from the real world, considering specific phenomena (Sprinkle et al., 2010). The same abstraction procedure can be applied in turn for the model itself. In that case, a so-called metamodel expresses certain properties of a model (Jeusfeld, 2009; Paige et al., 2014). Briefly, it could be said that "a meta-model is a model that consists of statements about a model" (Jeusfeld, 2009) or "a metamodel makes statements about what can be expressed in the valid models of a certain modelling language" (Seidewitz, 2003). Due to the fact that a wide range of definitions and standards (cf. Object Management Group or ISO/IEC 24744) concerning metamodels do exist (Atkinson and Kühne, 2003; Bézivin, 2004, 2005; Seidewitz, 2003), the understanding used within this paper shall be shared explicitly: "A metamodel is a description of the abstract syntax of a language, capturing its concepts and relationships, using modelling infrastructure" (Paige et al., 2014). The definition implies, that a metamodel just comprises the abstract and not the concrete syntax of a language, i.e. it lists the allowed constructs but it does not provide information about the right application (Jeusfeld, 2009). This circumstance allows for flexibility in the designing and deploying process (Paige et al., 2014). The main advantages of developing metamodels is that they document and support the language evolution over time, foster creation of well-formed models, support model-transformations, and formal checking of model properties (Paige et al., 2014).

3.1.2 Ontology based metamodel development

Gruber (1993) defines ontologies as a "specification of a representational vocabulary for a shared domain of discourse - definitions of classes, relations, functions, and other objects […]". Generally, metamodels can be understood as formalised ontologies (but not vice-versa). Given the variety of terms used for objects, relations, and attributes within the manufacturing domain, it is reasonable to make use of the more general guidelines for ontologies to prepare for metamodel development. This approach is used to capitalise on the advantages of thoroughly designed ontologies such as the explicit formulation of the structure of information and the underlying assumptions of a domain, enabling reuse of domain knowledge, and improving the quality of formal domain knowledge analysis (Noy and McGuinness, 2001). Using the ontology development guide of Noy and McGuinness (2001), steps 1 to 5 of have been carried out iteratively: (1) Determining the domain and scope, (2) searching for opportunities to reuse existing ontologies, (3) enumerating important terms for specified domain, (4) defining classes and the taxonomic hierarchy, and (5) defining the properties of classes. In order to support step 1 and 2, existing frameworks, taxonomies, and descriptions for categorizing factory objects, relations, and attributes - with a focus on German factory planning literature - have been analysed in detail.

3.2. Metamodel of nodes

The models depicted in Figure 2 and Figure 3Figure were designed according to the factory system definition formulated in section 1.2 and the requirements identified in section 2.1.

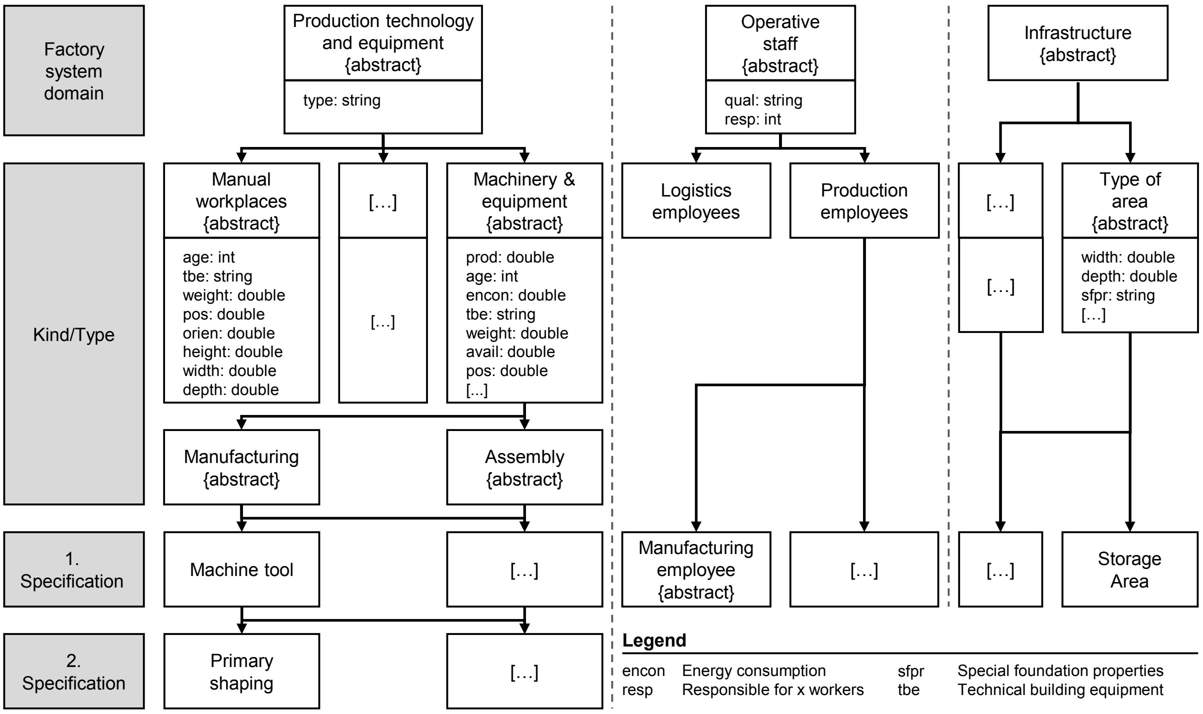

Figure 2. Developed nodes metamodel

The nodes metamodel in Figure 2 consists of four layers characterised by an increasing level of detail from top to bottom. Nevertheless, it is not mandatory for every branch to comprise all four layers. Each box in the model has an individual declaration such as 'Equipment' or 'Infrastructure'. Curly brackets including the term 'abstract', indicate that the node will not appear as concrete in the resulting model but is available only for the sake of inheritance. Besides, each class may contain attributes like it is the case for Equipment. Generally, attributes describe the properties of nodes, and edges are necessary to enrich the information content of the resulting model. Overall the developed metamodel consists of 71 classes and 20 attributes, in order to characterize factory objects and relations regarding performance, dimensions as well as physical and chemical properties. Due to the size of the model only an extract can be presented here.

3.3. Metamodel of edges

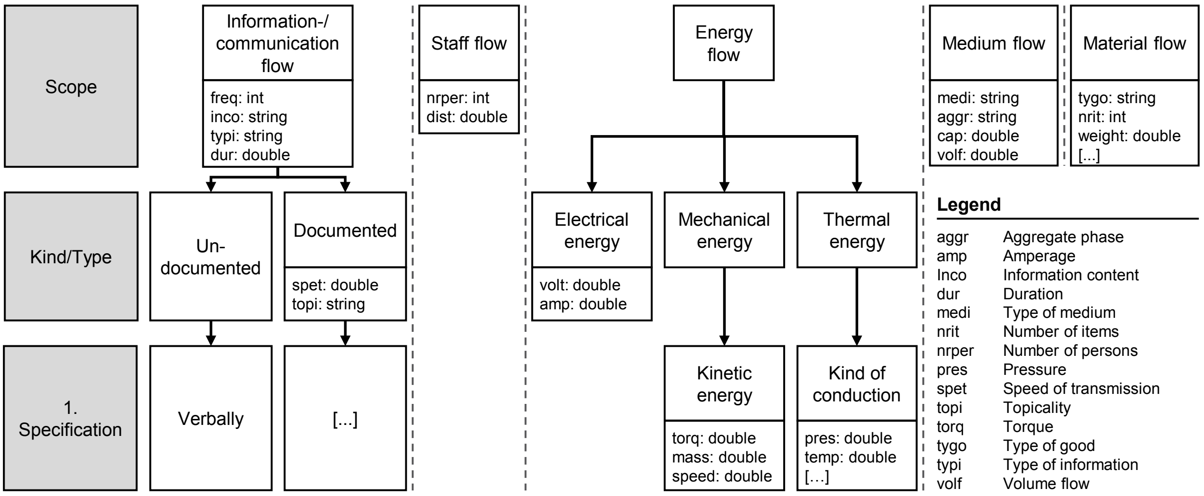

Figure 3. Developed edges metamodel

To represent the relationships between different factory objects, a metamodel of edges had to be designed as shown in Figure 3. In contrast to the metamodel of nodes, the edges metamodel has just three layers: 'Scope', 'Kind/Type' and '1. Specification'. It is assumed that the relationships between the objects can be described with five types of flows. Those again have an increasing level of detail from top to bottom. However, note that in contrast to the nodes metamodel all edges are classified as concrete. This way, the user is not obliged to specify the types of flows in more detail than the 'Scope' level. Classifying edges (or nodes) as abstract or concrete thus implicitly sets the level of detail within models.

4 APPLICATION

4.1. Academic example

The underlying academic example is excerpted from Müller and Ackermann (2013). In the original factory planning case study, five components of a compressor are produced using fifteen types of manufacturing equipment. The factory layout of this workshop production is illustrated in the upper part of Figure 4. In order to keep the applied example as simple as possible, the case study was reduced to just one component, which is a compressor-shaft, and six corresponding machine tools. Each of those stations is operated by a machinist or auxiliary worker. Besides the machine tools a bay warehouse, buffers, and several storage areas are required.

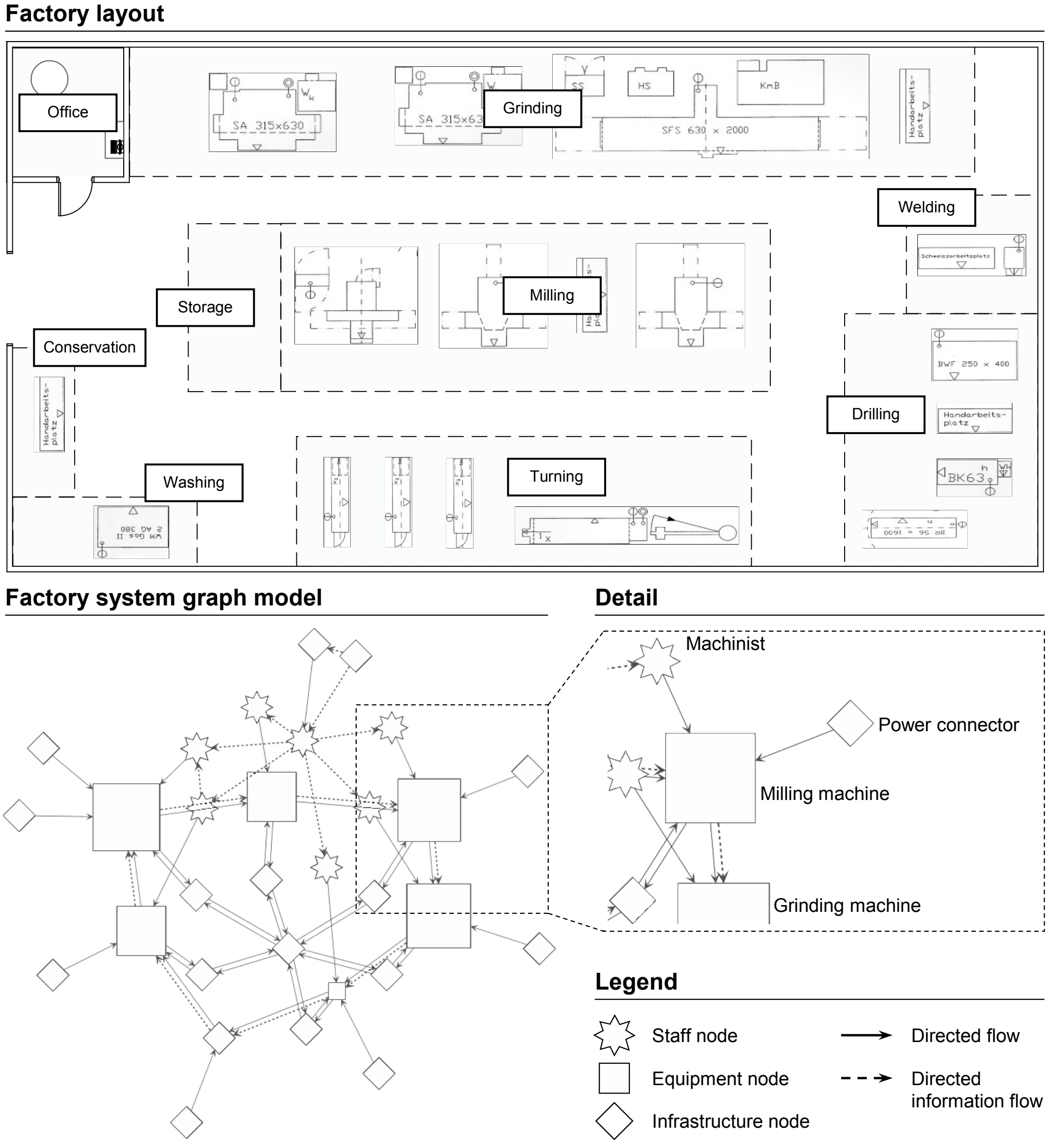

Figure 4. Exemplary graph model for a compressor shaft production

4.2. Design tool Soley Studio 2

The application example was modelled in Soley Studio 2 which is based on the work of Helms on object-oriented graph grammars (Helms, 2013). The general purpose of this software tool is to merge distributed data from different sources in a graph-based representation to support overarching analyses. Beside other features, Soley Studio 2 can detect hidden dependencies or patterns and is capable to model and visualise complex coherences in a clear, understandable way providing useful insights in extensive graph-based data. In addition, with its integrated workflow automation, analysis knowledge of domain experts can be formalized and applied through automated workflows. Thereby, the knowledge inherent in these workflows can be used by anyone, e.g. to perform sophisticated analyses in decision-making situations. For more information about the software please refer to: www.soley-technology.com.

4.3. Review of application experiences

Although the modelled factory system was strongly simplified Figure 4 shows a considerable level of complexity due to the amount of relationships among manufacturing resources. To exemplify a simple and intuitive visual analysis, all machine tools have been scaled linearly depending on their mass. The process of modelling revealed that the metamodel's level of detail is designed appropriate to capture important structural properties of the factory system according to the definition given in section 1.2 and taking reasonable design effort into account. However, the application example uncovered that some of the required information (e.g. the specific type of information or material flow) - unlike defined by the metamodels - can also be documented by attributes reducing modelling effort but not the information content of the model. Doing so allows the user to define customized attributes making the approach adaptable for a variety of application scenarios. Apart from this, the application exposed that some nodes possess properties which a user would normally expect to be related to edges. For instance, in the applied example a forklift is used for material handling between production stations. As the forklift is an object, it is currently modelled as a node but the material flow, accompanied by material handling between production stations, should be modelled with edges. A proposed solution for those cases (also applicable e.g. for pipes, cables or staff) would be to define that relational characteristics are treated as dominant. Future research needs to examine if this simplification reduces the analytical capabilities of the model.

5 CONCLUSION

5.1. Application perspectives and future research

The motivation for modelling factory systems explained in the introduction of this paper was to provide a basis for model-based analysis of ilities within complex socio-technical systems in the manufacturing domain. In this section, other promising application perspectives shall be proposed. As (Jarratt et al., 2011) point out, the assessment of change impacts in factory systems lacks suitable design support. The proposed metamodels could be used to analyse and predict mechanisms of change propagation depending on the system architecture and the respective type of manufacturing change considered analogous to Koh et al. (2012) and Koh et al. (2013). Changes in manufacturing systems are induced by a multitude of change causes such as external influences, engineering changes, and new production technologies. Further notable contributions within this field have been made by Giffin et al. (2009) and Hamraz et al. (2013).

Another potential field of application is the evaluation of the benefits of design for changeability (Fricke and Schulz, 2005) or flexibility in engineering design (de Neufville and Scholtes, 2011) on mitigating the undesirable effects of changes (e.g. time, effort, and cost) - or even on opportunities resulting from uncertainty - and to get further insights in the individual contribution of supporting design principles such as modularity, neutrality, and ideality. In the field of complexity management, which is a research topic of persistent interest in product development and engineering design literature, the metamodels presented here are helpful to support the transfer of existing matrix-based methods and to enrich the quality of analysis when applied to factory systems (cf. e.g. Lindemann et al., 2009).

Currently, the authors are studying opportunities to apply the modelling approach for a model-based evaluation of changeability-related factory system ilities. The extension of conventional graph-based modelling techniques (cf. section 2.2), understanding the interdependencies of relevant ilities, a meaningful interpretation of graph metrics, and the transfer of network analysis as well as visual analytics approaches are deemed as crucial steps on the pursuit of this research objective.

5.2. Summary

Uncertainty about future developments is one of the major concerns of manufacturing companies. Flexibility or changeability have proven to be beneficial system properties in a multitude of situations (de Neufville and Scholtes, 2011). However, analysing and quantifying these particular ilities is a challenging task because they are based on a variety of supporting principles which are at least partially interconnected and often of a rather qualitative nature. This paper tried to contribute to the model-based evaluation of these properties within the manufacturing domain by the design of a graph-based domain specific modelling approach comprising metamodels of nodes and edges including a variety of attributes as a first step in this research process. The design was based on metamodel and ontology design theory and an extensive analysis of existing frameworks in German factory planning literature. In addition, other promising fields of application have been outlined to inspire future research opportunities.

REFERENCES

Atkinson, C. and Kühne, T. (2003), "Model-Driven Development: A Metamodeling Foundation", IEEE Software, Vol. 20 No. 5, pp. 36–41.

Bézivin, J. (2004), "Model Engineering for Software Modernization", Proceedings of the 11th IEEE Working Conference on Reverse Engineering, p. 4.

Bézivin, J. (2005), "On the unification power of models", SoSyM, Vol. 4 No. 2, pp. 171–188.

Blessing, L.T.M. and Chakrabarti, A. (2009), DRM: A Design Research Methodology, Springer, London.

Chalupnik, M.J., Wynn, D.C. and Clarkson, P.J. (2013), "Comparison of ilities for protection against uncertainty in system design", Journal of Engineering Design, Vol. 24 No. 12, pp. 814–829.

de Neufville, R. and Scholtes, S. (2011), Flexibility in Engineering Design, Engineering Systems Series, MIT Press, Cambridge, MA.

de Toni, A. and Tonchia, S. (1998), "Manufacturing flexibility: a literature review", International Journal of Production Research, Vol. 36 No. 6, pp. 1587–1617.

de Weck, O.L., Roos, D. and Magee, C.L. (2011), Engineering Systems: Meeting Human Needs in a Complex Technological World, Engineering Systems Series, MIT Press, Cambridge, MA.

de Weck, O.L., Ross, A.M. and Rhodes, D.H. (2012), "Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)", 3rd International Engineering Systems Symposium (CESUN 2012), pp. 1–12.

Debbabi, M., Hassaïne, F., Jarraya, Y., Soeanu, A. and Alawneh, L. (2010), Verification and Validation in Systems Engineering: Assessing UML/SysML Design Models, Springer, Heidelberg et al.

Dori, D. (2002), Object-Process Methodology: A Holistic Systems Paradigm, Springer, Berlin Heidelberg.

Eppinger, S.D. and Browning, T.R. (2012), Design Structure Matrix Methods and Applications, Engineering Systems Series, MIT Press, Cambridge, MA.

France, R. and Rumpe, B. (2005), "Domain specific modeling", SoSyM, Vol. 4 No. 1, pp. 1–3.

Fricke, E., Gebhardt, B., Negele, H. and Igenbergs, E. (2000), "Coping with Changes: Causes, Findings, and Strategies", Systems Engineering, Vol. 3, pp. 169–179.

Fricke, E. and Schulz, A.P. (2005), "Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle", Systems Engineering, Vol. 8 No. 4, pp. 342–359.

Giachetti, G., Marín, B. and Pastor, O. (2009), "Using UML as a Domain-Specific Modeling Language: A Proposal for Automatic Generation of UML Profiles", in van Eck, P., Gordijn, J. and Wieringa, R. (Eds.), Advanced Information Systems Engineering, Lecture Notes in Computer Science, Vol. 5565, Springer Berlin Heidelberg, pp. 110-124.

Giffin, M., de Weck, O.L., Bounova, G.A., Keller, R., Eckert, C.M. and Clarkson, P.J. (2009), "Change Propagation Analysis in Complex Technical Systems", Journal of Mechanical Design, Vol. 131 No. 8.

Gruber, T.R. (1993), "A Translation Approach to Portable Ontology Specifications", Knowledge Acquisition, Vol. 5 No. 2, pp. 199–220.

Gunawan, I. (2012), "Managing Complex Engineering Projects with Design Structure Matrix Methods", in Mathew, J., Ma, L., Tan, A., Weijnen, M. and Lee, J. (Eds.), Engineering Asset Management and Infrastructure Sustainability, Springer London, pp. 275-282.

Hamraz, B., Hisarciklilar, O., Rahmani, K., Wynn, D.C., Thomson, V. and Clarkson, P.J. (2013), "Change prediction using interface data", Concurrent Engineering: Research and Applications, Vol. 21 No. 2, pp. 141–154.

Helms, B. (2013), "Object-Oriented Graph Grammars for Computational Design Synthesis", Diss., TUM, 2013.

Jarratt, T.A.W., Eckert, C.M., Caldwell, N.H.M. and Clarkson, P.J. (2011), "Engineering change: an overview and perspective on the literature", Journal of Research in Engineering Design, Vol. 22, pp. 103–124.

Jeusfeld, M.A. (2009), "Metamodel", in Liu, L. and Özsu, M.T. (Eds.), Encyclopedia of Database Systems, Springer US, pp. 1727-1730.

Koh, E.C.Y., Caldwell, N.H.M. and Clarkson, P.J. (2012), "A method to assess the effects of engineering change propagation", Journal of Research in Engineering Design, Vol. 23, pp. 329–351.

Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. and van Brussel, H. (1999), "Reconfigurable Manufacturing Systems", CIRP Annals - Manufacturing Technology, Vol. 48 No. 2, pp. pp. 527–540.

Lindemann, U., Maurer, M. and Braun, T. (2009), Structural Complexity Management: An Approach for the Field of Product Design, Springer, Berlin.

Maurer, M., Maisenbacher, S. and Kasperek, D. (2014), "Strukturbasierte Modellierung und Bewertung disziplinübergreifender Entwicklungszusammenhänge", in Vogel-Heuser, B., Lindemann, U. and Reinhart, G. (Eds.), Innovationsprozesse zyklenorientiert managen: Verzahnte Entwicklung von Produkt-Service-Systemen, Springer Vieweg, Berlin Heidelberg, pp. 15–29.

McManus, H.L., Richards, M.G., Ross, A.M. and Hastings, D.E. (2007), "A Framework for Incorporating "ilities" in Tradespace Studies", AIAA Space, Vol. 1, pp. 941–954.

Müller, E. and Ackermann, J. (2013), "Fallstudie Fabrikplanung. Lehrunterlagen zur Durchführung eines Planungsprojektes", available at: https://www.yumpu.com/de/document/view/27348097/fallstudie-fabrikplanung-technische-universitat-chemnitz/1 (accessed December 2014).

Noy, N.F. and McGuinness, D.L. (2001), Ontology Development 101: A guide to creating your first ontology, Stanford Knowledge Systems Laboratory Technical Reports.

Paige, R.F., Kolovos, D.S. and Polack, F.A.C. (2014), "A tutorial on metamodelling for grammar researchers", Science of Computer Programming, Vol. 96 No. 1, pp. 396–416.

Ross, A.M. (2008), Defining and Using the New "ilities", MIT Systems Engineering Advancement Research Initiative (Working Paper Series).

Ross, A.M., Rhodes, D.H. and Hastings, D.E. (2008), "Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining System Lifecycle Value", Systems Engineering, Vol. 11 No. 3, pp. 246–262.

Seidewitz, E. (2003), "What models mean", IEEE Software, Vol. 20 No. 5, pp. 26–32.

Sprinkle, J., Rumpe, B., Vangheluwe, H. and Karsai, G. (2010), "Metamodelling. State of the Art and Research Challenges", in Giese, H., Karsai, G., Lee, E., Rumpe, B. and Schätz, B. (Eds.), Model-Based Engineering of Embedded Real-Time Systems, Lecture Notes in Computer Science, Springer Berlin, pp. 57-76.

Steward, D.V. (1981), "The Design Structure System: A Method for Managing the Design of Complex Systems", IEEE Transactions on Engineering Management, Vol. 28 No. 3, pp. 71–74.

Valilai, O.F. and Houshmand, M. (2009), "Advantages of using SysML Compatible with ISO 10303-233 for Product Design and Development based on STEP Standard", Proceedings of the World Congress on Engineering and Computer Science, pp. 1–4.

Vokurka, R.J. and O'Leary-Kelly, S.W. (2000), "A review of empirical research on manufacturing flexibility", Journal of Operations Management, Vol. 18, pp. 485–501.

Wang, B., Madani, F., Wang, X., Wang, L. and White, C. (2014), "Design Structure Matrix", in Daim, T.U., Pizarro, M. and Talla, R. (Eds.), Planning and Roadmapping Technological Innovations, Innovation, Technology, and Knowledge Management, Springer International Publishing, pp. 53-65.

Wiendahl, H.-P., ElMaraghy, H.A., Nyhuis, P., Zäh, M.F., Wiendahl, H.-H., Duffie, N. and Brieke, M. (2007), "Changeable Manufacturing - Classification, Design and Operation", CIRP Annals - Manufacturing Technology, Vol. 56 No. 2, pp. pp. 783–809.

Wiendahl, H.-P. and Hernández, R. (2006), "The Transformable Factory - Strategies, Methods and Examples", in Dashchenko, A.I. (Ed.), Reconfigurable Manufacturing Systems and Transformable Factories, Springer, Berlin, Heidelberg, pp. 383–393.

ACKNOWLEDGMENTS

The German Research Foundation (DFG) funds this research and development project. We extend our sincere thanks to the DFG for the generous support of the work described in this paper, resulting from subproject B4 "Cycle-oriented production structure planning" in the framework of the Collaborative Research Centre 768 (SFB 768) "Managing cycles in innovation processes – Integrated development of product service systems based on technical products".