EXPLORATIVE APPLICATION OF THE MULTI-DOMAIN MATRIX METHODOLOGY IN LEAN DESIGN

Fabian A. Furtmeier[1], Iris D. Tommelein[2]

ABSTRACT

Structural complexity management provides a new approach to manage complexity resulting from a system's structure. It originated in the field of product development and applies the Multiple-Domain Matrix (MDM) methodology. The MDM methodology is based on the Design Structure Matrix (DSM) and focuses on the analysis and optimization of the underlying structures of a system. Therefore, it extends the capabilities of the DSM by integrating multiple domains and enabling the deduction of indirect dependencies. Dependencies that cannot be captured directly in a DSM can be computed by means of the MDM. Because construction projects are similar to extensive product development processes, methodologies developed for product development should be applicable in lean design in AEC industry.

Construction processes can be complex systems themselves. In order to avoid waste, these systems should be designed more accurate in the first place rather than improved while in operation. However, this requires process mapping tools that offer the capability to handle complex process networks. This paper explores the applicability of the MDM methodology as a process mapping tool in lean design. Therefore, the paper depicts the MDM methodology and illustrates an approach for mapping processes in lean construction by means of the MDM using the example of the design of a plumbing installation process. Objective of the MDM application is to enhance process mapping by use of deduction of dependencies. Further, conclusions for future state map generation are provided based on analysis of the process' underlying structure.

KEY WORDS

Complexity, Design structure matrix, Multi-domain matrix, Process mapping, Value stream mapping

INTRODUCTION

The application of lean thinking in the Architecture-Engineering-Construction (AEC) industry (namely lean construction) entails repercussions on complexity. For example, lean construction necessitates the early integration of all stakeholders. Especially in the design phase when objectives and value propositions are indistinct, the integration of all stakeholders increases a project's complexity because more dependencies have to be managed by the project team. Complexity is partly necessary to fulfill customer requirements and to deliver the project in time. Elimination of this complexity could make project delivery according to customer needs difficult. Consequently, complexity is not generally a hindrance to project success, but actually it can be a

---

[1] Dipl.-Ing., Graduate student, Institute of Product Development, Technical University of Munich, Germany, furtmeier@mytum.de

[2] Professor, Civil and Env. Engineering. Department, and Director of the Project Production Systems Laboratory (p2sl.berkeley.edu), 215-A McLaughlin Hall, Univ. of California, Berkeley, CA 94720-1712, USA, Phone +1 510/643-8678, FAX 510/643-8919, tommelein@ce.berkeley.edu

necessity for successful project delivery. In addition, complexity has been increasing within the AEC industry due to higher fragmentation of the industry and more complex building projects (Williams 1999).

Since construction is essentially an extensive product development process (Howell & Ballard 1994), lean construction shifts from a production view to an integrated view on all phases of the delivery process. Hence, construction processes should be designed accurately in the first place, instead of examined and modified during execution to fulfil delivery more efficient. This requires establishing a realistic picture of the process during the design phase when value propositions and delivery are still not fully specified. An additional challenge is that processes are often branched and complex. Process mapping tools that are applied for process design need to cope with the partial process state and should provide the capability to capture the interconnectedness of a process' steps.

MOTIVATION & HYPOTHESIS

Value stream mapping (VSM) complex processes can help to manage waste and value, but most current tools are not practical for depicting the branched and iterative nature of a complex process. A reason for this is that VSM illustrates the flow of a product and highlights wasteful steps but does not aim at capturing complexity. Several process mapping tools, e.g., cross-functional process mapping (Damelio 1996), offer the capability to capture network-like processes, but most tools do not aim at capturing the branched nature of value streams in one map but require multiple ones (Rother & Shook 1998, König et al. 2008, McManus 2005, Millard 2001). Processes and their tasks are often highly dependent on each other what makes capturing the underlying structure of the overall process right away difficult. Separate analysis of multiple maps is impractical as tasks are often highly interconnected.

Current tools do not regard complexity as an attribute of a process that arises from the process' underlying structure and structural characteristics (Lindemann et al. 2009). However, the development of a future state map, which is required to enhance the efficiency of a process, implies a sound understanding of the process' underlying structure. Nevertheless, missing knowledge of dependencies constrains effective future state map generation. However, in-depth analysis of this structure calls for a tool capable of illustrating the process' structure. Consequently, this paper proposes the following hypothesis: *"The application of the MDM as a process mapping tool can extend current capabilities of VSM tools, because the MDM allows for a more comprehensive access to a system's underlying structure."*

THE MULTI-DOMAIN MATRIX (MDM)

THE CONSTRUCTION OF THE MDM

A MDM represents all matrices that are required for storing information on a system's elements and their links in one single matrix system. DSM and Domain-Mapping Matrix (DMM) are the two central elements of a MDM. A DSM as an intra-domain matrix represents the dependencies within one domain. A DMM as an inter-domain matrix represents the dependencies between two different domains. A domain is defined as any point of view on a complex system (e.g., a product's components, requirements, people or processes). In contrast to other matrix-based approaches that combine intra- and inter-domain matrices, a MDM allows for the computation of unknown matrices (Lindemann et al. 2009). Consequently, a MDM can enhance system modelling because subsets showing indirect dependencies within a domain can be derived by information stored in other subsets. Figure illustrates a MDM and its items.

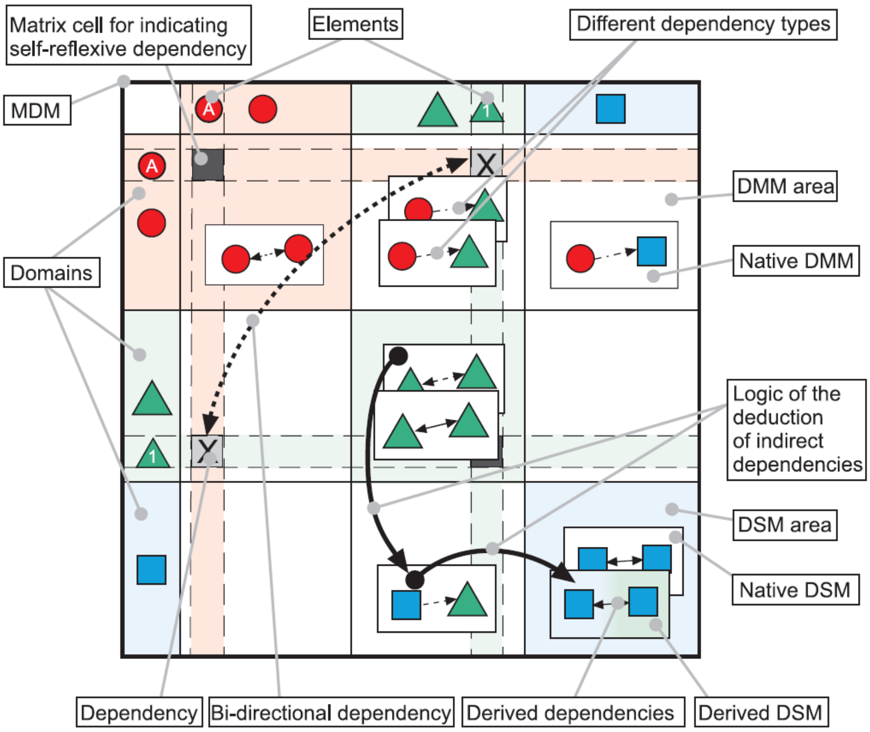


Figure 1: Items of the MDM (Maurer 2007)

DSMs are compromised along the MDMs diagonal while DMMs are arranged outside the diagonal. The matrix illustrated in Figure 1 has three different domains (indicated with different symbols: circle, square and triangle). Respectively, the MDM includes three DSMs which are located at the intersection of the same domains. A MDM contains a minimum of two domains but can be extended to an infinite number of domains. An element of a domain represents a column and a row in the MDM. The intersection of column and row consequently represent the element's self-reflexive dependency. Every other intersection of column and row represents a potential dependency between these elements. A mark or number is set in case of existence of a dependency. Symmetrically marks show bi-directional dependencies between two elements. Generally, a MDM is not a two-dimensional matrix as some domains have more than one type of dependency. Components of a product for example can have functional dependencies as well as geometrical dependencies. The dependency type characterizes the meaning of the dependency. Each DSM or DMM represents only one dependency type. Consequently, separate matrices have to be created within the same DSM or DMM area. DSMs and DMMs are named "native" if they result from data acquisition (e.g., interviews or surveys). "Derived DSMs" are computed through logics of deduction which are described in the next subsection.

DEDUCTION OF DEPENDENCIES

The computation of indirect dependencies is the most distinct feature of the MDM (Maurer & Lindemann 2007). Indirect dependencies between two elements result from dependency chains formed by the acquired direct ones (Lindemann et al. 2009). Figure 2 illustrates an example of an indirect dependency as a result of two direct ones. Understanding the logics of deduction of indirect dependencies is central for the definition of the system to be described by the matrix and consequently for the application of the MDM methodology because indirect dependencies and missing matrices can be deduced by the means of the computational logics.

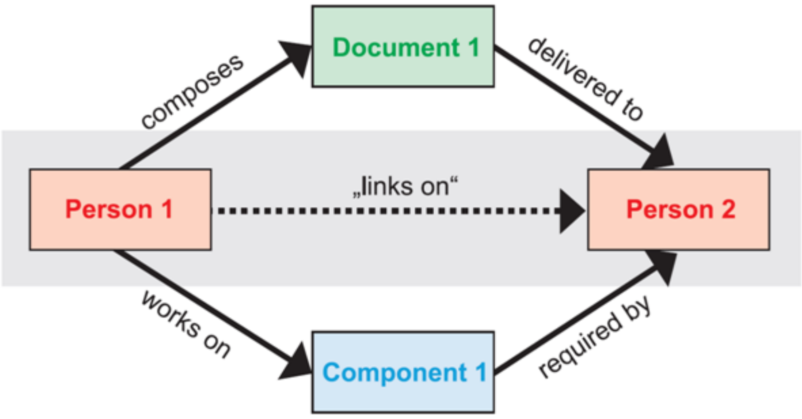


Figure 4: Indirect dependency (Lindemann et al. 2009)

If the direct dependencies possess a direction, six different logics exist for determining an intra-domain network from dependencies between two different domains. Consideration of undirected and bi-directional dependency types would increase the quantity of derivable DSMs but the logic procedure remains the same (Maurer 2007). Figure introduces the six basic computational logics briefly.

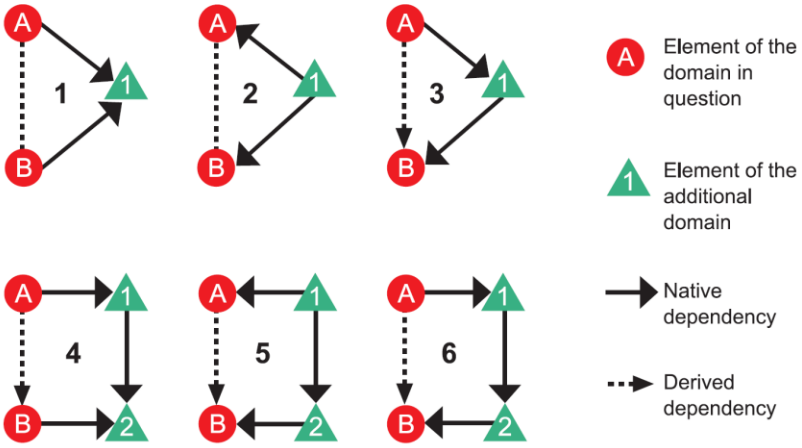


Figure 3: Logics for deriving indirect dependencies (Lindemann et al. 2009)

The presented deduction logics provide tremendous possibilities for the modelling and analysis of complex systems since the deduction of DSMs offers a methodical access to specific system views (Maurer 2007). The understanding of this feature of the MDM methodology is relevant for the successful application of the MDM in structural complexity management. For a better understanding of the deduction logics the following example is used: indirect dependencies between people (circles) are derived by existing (acquired) dependencies of people to documents (triangles). The example is a typical scenario in product development where designers depend on other designers due to mutual data transfer (Maurer 2007). In this case, the deduction of the designers' dependencies captures a more realistic perspective of these dependencies than acquiring the dependencies directly since the designers often do not know who requires a certain document next (Kusiak 2008, Maurer 2007).

CASE STUDY: MDM AS A PROCESS MAPPING TOOL

The case demonstrates the application of the MDM as process mapping tool in the design of the installation process for the plumbing in patient rooms in a hospital. The responsible trade partner already developed a conceptual plumbing installation plan based on experience. Application of the MDM as a process mapping tool allows a profound understanding of the process' structure. Conclusions, based on the knowledge of the structural characteristics of the process, provide the fundamental basis for the development of a future state map.

PROBLEM DESCRIPTION

Since a conceptual plumbing installation plan was already developed, main purpose of the MDM application was to enhance the current state plan and develop a production plan to most efficiently deliver plumbing services for the hospital. Structure analysis can provide the basis for the development of this future state process. In a first step, the process development team laid out the installation tasks for the plumbing system in a cross-functional process map. However, the cross-functional process map does not present any dependencies within the tasks (Damelio 1996). Development of a current state representation in a VSM turned out to be difficult because more than one material flow existed. Complexity arises because the process covers the installation of different systems in the patient room. Multiple workers will conduct the work and just-in-time material supply is planned to reduce waste in the project. Because it is a branched process and its tasks are highly dependent on each other, capturing the underlying structure of the overall installation process right away is difficult. Furthermore, a separate analysis of each system's installation process is impractical as the processes are highly interconnected.

Capturing processes by drawing arrows and boxes in a graphical representation increases difficulties for the users to indentify if all possible dependencies have been considered (Lindemann et al. 2009). Consequently, mapping complex and interconnected processes with conventional tools can be rather difficult (McManus 2005, Millard 2001). At this point, future state map generation can be advanced by an extension to the current process mapping and VSM tools by a methodology that provides the deduction of the dependencies between process steps and supports structure analysis. The challenge is to map the process accurately in a DSM. Because dependencies are difficult to capture, the DSM cannot be directly acquired in an accurate state. However, the MDM provides systematic access to capturing all dependencies and makes the underlying structure accessible.

FRAMEWORK OF MODELLING METHOD

Accurate acquisition of the dependencies between the tasks of a process is the major difficulty in mapping complex processes. Deubzer & Lindemann (2008) illustrate an application of the MDM for functional modelling (Ehrlenspiel 2003). The linkages between operations and states and vice versa are presented in two DMMs. Next, the dependencies between operations or states can be derived by means of the deduction logics as presented by Maurer (2007, see case 3 in Figure). Tasks are connected to their inputs and outputs in a similar way than operations and states are connected. Hence, this modelling approach can be applied to process mapping. Inputs and outputs in production processes are usually physical material inventories. Because of the flow of information in product development processes, inventories exist usually in form of documents, drawings or databases (Austin et al. 2000; Browning 2001; Eppinger 2001). In both cases, a certain process step, as a set of actions which accomplishes a certain assignment, transfers the subject of consideration from an input to an output, while changing the subject's attributes. Consequently, input inventories deliver to tasks and tasks deliver to output inventories which are input inventories for the next process steps. The inventories can also be considered as the state of the subject of consideration between two process steps. Hence, the notation is the same as for functional modelling. Consequently, the same logic of deduction derives the dependencies between the tasks. The resulting network of dependencies presents the flow within the tasks. For construction operations, this flow would be material flow. In product development or lean design processes, this flow would be information flow. Figure illustrates the modelling approach.

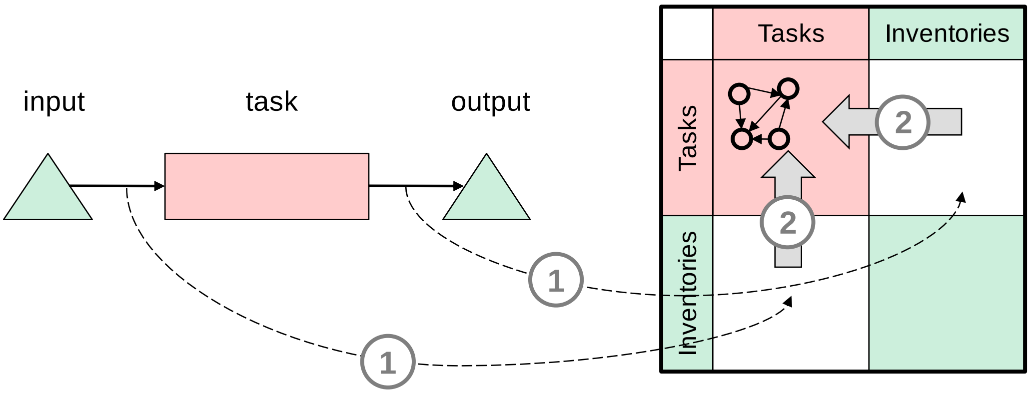


Figure 4: Modelling Framework (adapted from Deubzer & Lindemann 2008)

The cross-functional process map provided the information for the acquisition of the two native DMMs connecting inventories and tasks. First, a list of tasks was gathered from the cross-functional process map. Next, input and output inventories were allocated to the tasks. With the information on hand, the DSM containing the network of tasks can be derived by means of the applicable deduction logic. This DSM was computed from information stored in the two opposite DMMs linking the tasks with the inventories.

STRUCTURE ANALYSIS

Structure analysis comprises the identification of a system's characteristics and the derivation of calls for action. The identification of characteristics helps to develop a substantial understanding of the system in question. This knowledge makes the system behaviour more predictable and planning changes of the current state process is improved accordingly. The derived network of process steps can be visualized as a graph and as a matrix. Analysis of both forms of process' representation expands from focusing on the characterization of the entire structure and its behaviour to a specific focus on the structural embedding of single elements and dependencies. The objective of the MDM application is the identification of the root causes of problems and the conclusion of potentials for the system's optimization. It is important that analysis objective are planned exactly in order to avoid unnecessary or confusing analysis. The analysis of the derived network of tasks necessitates suitable analysis criteria for the characterization of the edges and nodes, subsets, and the entire network that allow an interpretation from a VSM point of view. In this case, edges represent the process' tasks and nodes illustrate the dependencies between those tasks. A subset is group of tasks which are closely related. Lindemann et al. (2009) provide basic analysis criteria for the classification of nodes and edges, subsets, and systems. These analysis criteria enable process analysis and have to be interpreted from a lean thinking point of view.

The deduced network was analyzed by means of certain criteria. Several structural characteristics were identified that can help to improve the current state map. Figure shows the deduced network of tasks and pinpoints structural characteristics.

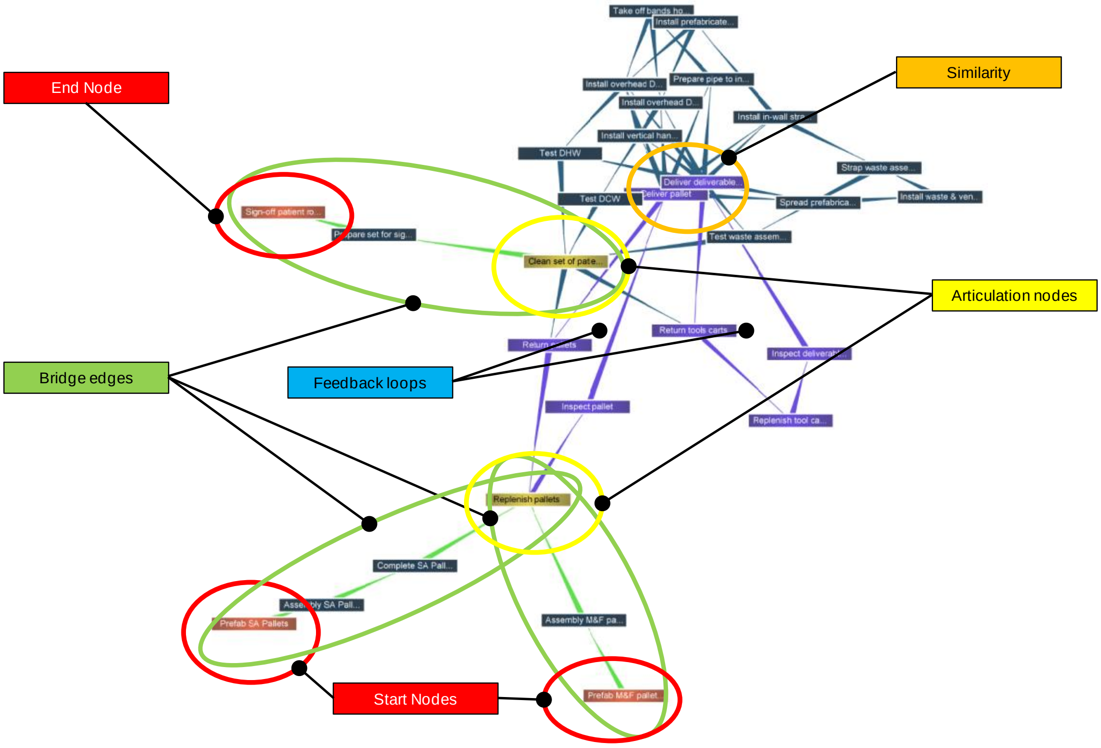
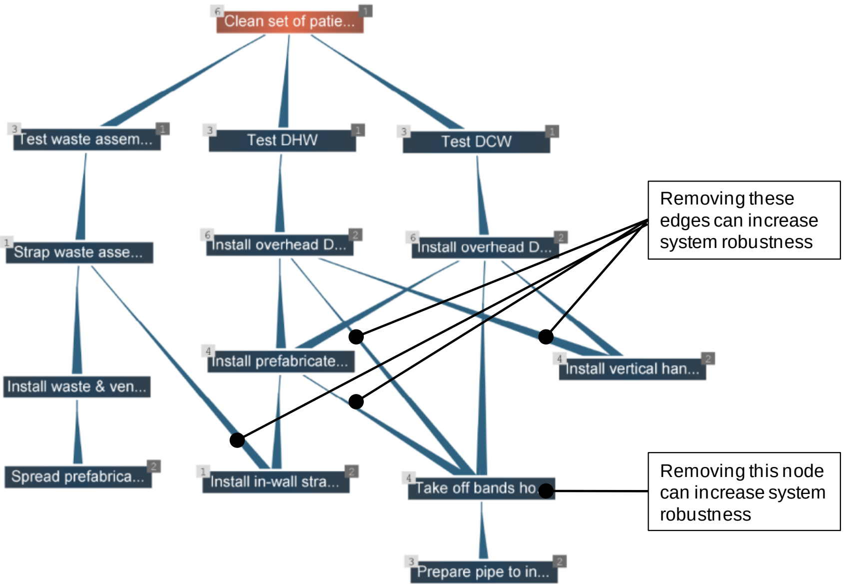


Figure 5: Deduced network of tasks

The deduced structure shows a highly interconnected subset and three sequential (bridge edges) paths connecting the subset with the end and start nodes. Articulation nodes connect these paths to the subset in the middle. The sequential paths can be seen as separate processes delivering to the subset, or receiving from the subset. They are connected by articulation nodes which form bottlenecks and therefore can define the processes' takt-time. Two feedback loops exist in the structure. In this case, these loops show the opportunity for KANBAN supply of materials. As shown in Figure 5, the left feedback loop pulls material out of the two paths. Furthermore, the structure shows a similarity. This similarity points out two tasks that can be integrated in one task. As those tasks are both part of the feedback loop, it might be possible to integrate the feedback loops as well.

Moreover, the structure shows a hierarchy, beginning at the articulation node and connecting the end node with the subset in the middle. The hierarchy illuminates the material flows converging in the articulation node. Three separate material flows can be identified. By removing the edges between these three flows, the installation process robustness and flexibility can be increased because the three flows no longer depend on each other. Figure 6 illustrates the hierarchy.



Removing these edges can increase system robustness

Removing this node can increase system robustness

Figure 6: Hierarchy illustrating material flows

Because the deduced network consists of almost only feed-forward processes, the matrix-based approaches clustering and triangularization do not provide improvements.

DISCUSSION OF PRACTICES

The application of the MDM methodology enables the user to map out a process by acquiring information on how the tasks within a process are connected to their respective inputs and outputs (inventories). Hence, a DSM containing the dependencies between the tasks can be derived by means of deduction logic. Structural complexity management provides criteria for the characterization of the process' structure. Analysis of these characteristics leads to suggestions for improvement of the process. The case study showed that these criteria can be interpreted from a lean thinking perspective. This combination of lean thinking principles, such as value and waste, and structural characteristics support VSM because MDM application enables the deduction of dependencies which helps to fetch a more comprehensive picture of a process than solely drawing a process map. This facilitates a more comprehensive understanding of the process and therefore can lead to better process improvement. In example, the hierarchy illustrates different possibility for work structuring which could reduce the projects takt-time to half a day instead of one day. Analysis of feedback loops allowed for an interpretation of iterations and validated the application of KANBAN.

In this case, the deduction of dependencies seems complicated in comparison to the “simple” process it was applied to. However, the MDM application significantly facilitated mapping out the process and achieved a level of detail that would have not been reached without MDM application.

CONCLUSIONS

The MDM application allowed for the identification of complexity deriving from a process' structure. However, the case shows only a few advantages of the MDM for managing complexity in general and as a process mapping tool in specific. The application extends the MDM as a process modelling tool (König et al. 2009) in the area of lean thinking and facilitates process improvements due to the application of structure analysis. Since Tuholski & Tommelein (2009) already successfully applied the DSM to analyse design iteration, the application of the MDM to map out design processes seems to be the next promising step. The uncertain and often unstructured nature of design processes causes difficulties with DSM application. However, application of the MDM may offer a means to capture and analyze the branched and complex nature of design processes due its distinct features and consequently should be further investigated.

In addition to the similarities between lean construction and lean product development, the circumstance that project delivery can be seen as a system (Bertelsen 2002) provides another reason for further investigation of structural complexity management in lean construction. Regarding complexity from a lean construction point of view, it can neither be seen as waste nor value per se, but it should be seen as a system's characteristic leading to and resulting from waste or value. Sole elimination of complexity cannot be a valid strategy since complexity can relate to value. Thus, complexity management in lean construction needs to distinguish complexity linked to value from complexity linked to waste. It should focus on the active management of complexity linked to value and the elimination of complexity linked to waste. As stated, structural complexity management can provide systematic access to a system's and consequently help to make this distinction. However, further investigations are necessary to ascertain applicability and practicality of structural complexity management and to evaluate its relevance for improving project delivery.

ACKNOWLEDGMENTS

Thanks are due to the Cathedral Hill Hospital Project Team and especially Ben Tapparo and Mark Blackwell from Southland Industries for their valuable contributions and insights.

REFERENCES

Austin, S. A.; Baldwin, A.; Li, B.; Waskett, P. (2000). „Application of the Analytical Design Planning Technique to Construction Project Management". *Project Management Journal* 31 (2), pp. 48-59.

Bertelsen, S. (2002). "Complexity - Construction in a New Perspective." *Proceedings of the 10th Annual Conference of the International Group for Lean Construction (IGLC-10)*. IGLC-10, Gramado.

Boardman, J., and Sauser, B. (2008). *Systems Thinking: Coping with 21st Century Problems*. CRC Press, Boca Raton.

Browning, T. R. (2001). "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions." *IEE Transactions on Engineering Management* 48 (3), pp. 292-306.

Damelio, R. (1996). *The Basics of Process Mapping*. Productivity Inc., Portland.

Deubzner, F.; Lindemann, U. (2008). "Functional Modelling for Design Synthesis using MDM Methodology." In: Kreimeyer, M.; Lindemann, U.; Danilovic, M. (Eds.), *Proceedings of the 10th International Design Structure Matrix (DSM) Conference, Stockholm*. Hanser, Munich, 403-411.

Ehrlenspiel, K. (2003). *Integrierte Produktentwicklung*. Carl-Hanser, Munich.

Eppinger, S. D. (2001). “Innovation at the Speed of Information.” *Harvard Business Review* 79 (1), pp. 149-158.

Howell, G. A.; Ballard, G. (1994). “Lean Production Theory: Moving Beyond 'Can-Do'.” In: Alarcon, L. (Ed.), *Lean Construction*. A.A. Balkema, Rotterdam, 17-23.

König, C.; Kreimeyer, M.; Braun, T. (2008). „Multiple-Domain Matrices as a Framework for Systematic Process Analysis.” In: Kreimeyer, M.; Lindemann, U.; Danilovic, M. (Eds.), *Proceedings of the 10th International Design Structure Matrix (DSM) Conference, Stockholm*. Hanser, Munich, 231-244.

Kusiak, A. (2008). Interface Structure Matrix for Analysis of Products and Processes. In: *Proceedings of the 15th CIRP International Conference on Life Cycle Engineering (LCE 2008)*. The University of New South Wales 2008, Sydney, 444-448.

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

Maurer, M. S. (2007). *Structural Awareness in Complex Product Design.* TU Munich, Chair of product development, Dissertation.

Maurer, M. S.; Lindemann, U. (2007). *Facing Multi-Domain Complexity in Product Development*. Cidad Working Paper Series 3 (2007) 1, 1-12.

McManus, H. L. (2005). *Product Development Value Stream Mapping (PDVSM) Manual 1.0*. Lean Aerospace Initiative Center for Technology, Policy, and Industrial Development, MIT, Cambridge, USA.

Millard, R. L. (2001). *Value Stream Analysis and Mapping for Product Development.* MIT, Master Thesis, Cambridge, USA.

Rother, M.; Shook, J. (1998). *Learning to See. Value Stream Mapping to Add Value and Eliminate Muda.* The Lean Enterprise Institute, Brookline.

Williams, T. M. (1999). “The Need for New Paradigms for Complex Projects.” *International Journal of Project Managemen.* 17 (5), 269-273.