A. M. Heinecke, H. Paul (Hrsg.): Mensch & Computer 2006: Mensch und Computer im Struktur*Wandel*.
München, Oldenbourg Verlag, 2006, S. 379-384

Towards Tangible Work Modelling

Stefan Oppl, Christian Stary, Andreas Auinger

Johannes Kepler Universität Linz

Abstract

Digital Montessori-inspired Manipulatives (MiMs), so far have been used effectively for knowledge transfer in elementary educational settings. For work modelling and task-based interactive systems design we propose a modelling concept and usage scenario that should help to increase effectiveness in organization and technology development. The Tangible Task Modelling Demonstrator (TTMD) facilitates the representation and development of work and task models by means of MiMs. Using the TTMD users can directly grasp and manipulate work tasks. As a result, the cognitive load for modelling and (re-)arranging elements of work spaces can be reduced.

1 Motivation

Interactive work models represent and keep the alignment between multiple elements of an organization. This type of modeling is not only fundamental to understand how an organization operates and adapts to changing business environments (cf. Caetano et al. 2005), but also to design task-centered user interfaces (cf. the recent series of ACM TAMODIA conferences). Although the modeling process facilitates human understanding and communication, it is by no means clear what kind of notation should be used to facilitate the communication between stakeholders, business specialists, and interactive software developers (cf. Holmboe 2005; Oppl et al. 2005). Nevertheless, identifying the properties and relationships of work tasks is fundamental to help understanding and evolving interactive work support.

The requirement for proper representation techniques as a mediator between technology-oriented people and workers has already been recognized. In the field of Semantic Web Dori (2004) points out that 'humans and machines must each use a different format of knowledge representation' (p.121). In order to reconcile the apparent human-machine language orientation dilemma, not only the modeling process, but also the representation of design knowledge has to be revisited, since the latter is a critical (cognitive) concept (Crapo et al. 2000).

Both, the modeling process, and the domain knowledge representation with respect to work tasks, can be addressed through adequate concepts for visualization and hands-on support, as recent studies with Digital Montessori-inspired Manipulatives (MiMs) (cf. Zuckerman et al. 2005) reveal. It also facilitates capturing the intended semantics of work domains – a feature that currently cannot be provided even by standardized specification approaches (cf. Harel &

Rumpe 2004 for UML). In this paper we focus on work-task and business process modeling according to different aspects, of which each can then be handled independently and later composed to synthesize a comprehensive model of a work domain. To do so, we propose defining two complementary representations, a direct manipulative tangible and a conventional UML-based one. We argue that using the tangible objects to model work tasks and business processes improves the understandability of the individual work objects and of the adjunct business process, in line with the successful hands-on organization of discrete pieces of abstract information (cf. Jacob et al. 2002), and discrete physical object manipulation representing digital information (cf. Ullmer et al. 2005). The Tangible Task Modeling Demonstrator (TTMD) builds upon approaches like Task Blocks (Terry 2001), a system that uses physical blocks (task blocks) to represent computational functions. In our case not only "pipelines" can be created that sequentially manipulates data, but also entire business process representations.

2 The Tangible Task Modeling Demonstrator

The dual presentation approach of the Tangible Task Modeling Demonstrator (TTMD) entails several implications for the modeling process. Using a cluster case study from the automotive sector (see also URL: http://www.CrossWork.org), the basic steps of modeling are described in the following. In the case of networked organizations which work on a common customer order the organization of work has to be revisited from case-to-case, according to the capabilities of the cluster partners. We exemplify an enterprise in the automotive sector that has to analyze its supplier selection process, e.g., to handle specific orders. For clarification on how to handle a particular order the CIO arranges a meeting involving the heads of the involved departments, the supplier manager and the product manager. Confronted with a blank modeling surface, the group starts to model the initial activity, e.g., 'specification of goals' by placing a respective block and labeling it. After a short discussion they add the role responsible for executing that activity, and the data resulting from this activity (see figure 1).

Figure 1: Initial Modeling Steps

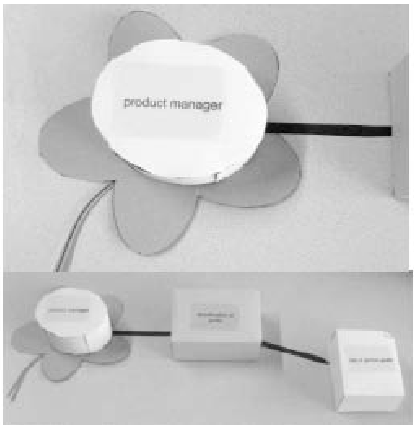

Figure 2: Using a Contextual Symbol

The involved heads of department recognize the product manager not being the only one involved in the goal specification activity. It is rather a cooperative activity involving all department members. Since there are five different departments consulted by the product manager, they term this involvement 'the shamrock' with each leaf representing a department. Accordingly, they define a corresponding contextual information symbol. It shadows the role 'product manager' (see figure 2).

They continue modeling by placing another activity – 'goal decomposition' – requiring the data produced by the first activity as an input. Since that activity is handled by the product manager in the same way as the activity before (again involving the shamrock), they connect the respective role to this activity. As a new output, a data block labeled 'list of parts' is placed (see figure 3).

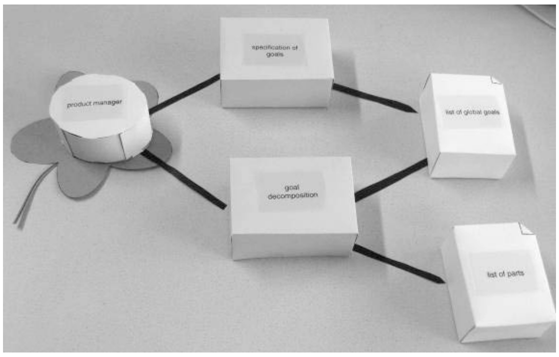

Figure 3: Modeling Interdepencies between Process Elements

While most of the involved persons feel this placement to fit their view on the process, the supplier manager is not satisfied. He would like to formalize this activity based on a set of textual instructions that have to be applied in any case. After a short discussion, they decide to include this information in the model. They use a 'Data'-Artifact to be put into the concerned 'Activity'-Block (now used as a container). To do so, initially they need to dock the digital instruction document to the artifact via an explicit binding interface of the TTMD. Subsequently, the artifact can be put into the activity container physically (see figure 4).

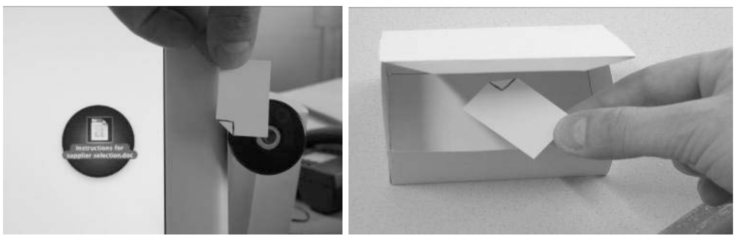

Figure 4: Use of Artifacts for Element Refinement

They finish modeling the process by connecting the activity to the input data block and assigning a role for execution, in this case ‘supplier manager’. As the shamrock is not involved in this activity, this role has not to be shadowed.

Since the modeling process has been tracked permanently in an unobtrusive way by the TTMD, immediately after polling the physical surface data the model becomes available on the virtual interface for future reference or reuse. The CIO marks this version of the model to be the final one. The captured history of creating the model also becomes available. This feature is especially useful to convey the rationale of modeling and to step back to different and/or previous design versions at the virtual level (see figure 5 for the dual interfaces).

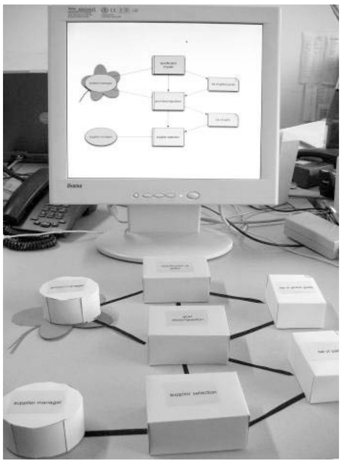

Figure 5: Dual View of Work Processes

Dual View While traditional modeling approaches basically assemble work processes of activities, roles and data, the TTMD modeling language provide individualized and visual means to capture contextual information concerning the organization of work. In this way, the intelligibility can be increased for all parties (cf. Oppl et al. 2005).

Based on the developed diagrammatic notation scheme and its counterpart for interactive hands-on modeling the TTMD software and hardware system enabling the creation and manipulation of work models has been designed. The core of such an environment has to handle the dual view (physical ←→ computer-based) and the physical modeling blocks to create task models including both their organizational and processing context. Correspondingly, the TTMD is composed of two parts, a physical and a virtual ‘world’ management, with some functionality available in both and some of it distributed.

While in most cases the interfaces between virtual and physical presentation layers have to be transparent for the users, there are certain cases in which the explicit transfer of information from the virtual part to the physical surface and vice versa is required. Those cases require a dedicated management facility. Accordingly, we distinguish two intertwined components of the TTMD: the physical surface and the virtual surface. These surfaces are linked through implicit interfaces (which are not visible for users) and explicit interfaces (which require explicit interaction with the user) (see figure 6).

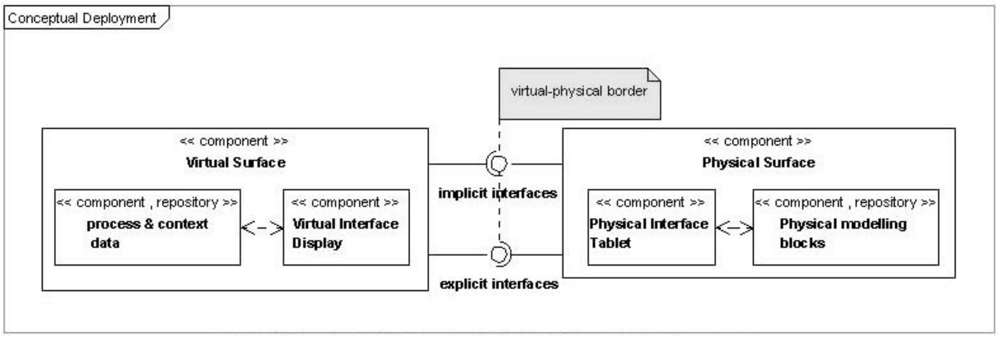

Figure 6: Conceptual Overview

The surface parts are used as follows: The actual modeling of a process is carried out on the physical surface. The virtual surface can also be used for modeling and is additionally able to export or archive existing process models and their development steps. The two surfaces are permanently synchronized via the implicit interfaces, e.g., the underlying sensor infrastructure. Data that are available initially on the virtual surface but required on the physical surface, such as newly defined contextual symbols, are made available physically through explicit interfaces.

The TTMD also provides proper hardware support for modeling as described above. Although both, a straightforward *Smart Thing* approach (with IT-infrastructure merely in the manipulatives), and a corresponding *Smart Space* approach (with IT-infrastructure only in the physical surface) can be pursued for implementation, existing solutions (Jacob et al. 2002; Patten et al. 2001) show that a combined approach increases the degrees of freedom (cf. Oppl 2006).

In order to facilitate the communication between the virtual surface and the physical surface, a standardized data representation for the software interface is crucial. Methodologies and notations for process descriptions, such as ARIS (Scheer 2003), and task modeling, such as UML (Booch et al. 1999) use the XML metadata representation language XMI (XML Metadata Interchange proposed by OMG 2006). It facilitates the storage and compatible interchange of task and process representations. The XMI-compliant TTMD-representation scheme has been generated using the Eclipse Modeling Framework. The XMI scheme as a container for concrete data is used by the physical surface. It tunes functional elements with the software required for the virtual presentation of the work models. All XMI structured data are stored in the process- and context-data repository (see figure 6).

The TTMD software support system has been designed

1. to mirror the processes modeled on the physical surface at a virtual one
2. to provide extensive modeling capabilities addressing contextual information

3. to store and retrieve modeling data, its creation and changes
4. to output data to the physical surface via implicit or explicit interfaces.

In this way, we were able to enhance Digital Montessori-inspired Manipulatives for interactive system design including work tasks.

References

Booch, G.; Rumbaugh, J.; Jacobson, I. (1999): The Unified Modeling Language User Guide. Addison Wesley.

Caetaono, A.; Silva, A.R.; Tribolet, J. (2005): Using Roles and Business Objects to Model and Understand Business Processes. Proc. SAC'05, ACM, pp. 1308-1313.

Crapo, A.W.; Waisel, L.B.; Wallace, W.A.; Willemain, Th. R. (2000): Visualization and the Process of Modeling: A Cognitive-theoretic View. Proc. KDD'00, ACM, pp.218-226.

Dori, D. (2004): ViSWeb – The Visual Semantic Web: Unifying Human and Machine Knowledge Representations with Object-Process Methodology. VLDB-Journal 13, pp. 120-147.

Harel, D.; Rumpe, B. (2004): Meaningful Modeling: What's the Semantics of "Semantics"? IEEE Computer 37(10), pp. 65-72.

Holmboe, Ch. (2005): The Linguistics of Object-Oriented Design: Implications for Teaching. Proc. ITiCSE'05, ACM, pp. 188-192.

Jacob, R. J. K.; Ishii, H.; Pangaro, G.; Patten, J. (2002): Hands-On Interfaces: A Tangible Interface for Organizing Information Using a Grid. Proc. CHI'02, ACM, 2002.

Montessori, M. (2005): The Montessori Method, Kessinger Publishing.

Object Management Group (2006): MOF 2.0, XMI Mapping Specification 2.1. http://www.omg.org /technology/documents/formal/xmi.htm (last visited 27.02.2006).

Oppl, S. (2006): Towards Intuitive Work Modeling with a Tangible Collaboration Interface Approach, Proc. WETICE 2006 (TICE 2006), IEEE, 2006.

Oppl, S.; Stary, Ch. (2005): Towards Human-Centered Design of Diagrammatic Representation Schemes. Proc. TAMODIA'05, ACM, 2005.

Patten, J.; Ishii, H.; Hines, J.; Pangaro, G. (2001): Sensetable: A Wireless Object Tracking Platform for Tangible User Interfaces. Proc. CHI 2001, ACM, pp. 253-260.

Scheer, A.-W. (2003): ARIS – Business Process Modeling. Springer, Berlin.

Terry, M. (2001): Task Blocks: Tangible Interfaces for Creative Exploration. Proc. CHI, ACM.

Ullmer, B; Ishii, H.; Jacob, R. J. K. (2005): Token+Constraint Systems for Tangible Interaction with Digital Information. ACM-TOCHI 12.

Zuckerman, O.; Arida, S.; Resnick, M. (2005): Extending Tangible Interfaces for Education: Digital Montessori-inspired Manipulatives. Proc. CHI, ACM, pp. 859-868.

Kontaktinformationen

Institut für Wirtschaftsinformatik – Communications Engineering, www.ce.jku.at