Design of a suite of visual languages for supply chain specification

Rick Zhang[1], John Hosking[1], and John Grundy[2], Nikolay Mehandjiev[3] and Martin Carpenter[3]

[1]Department of Computer Science, The University of Auckland, New Zealand

[2] Computer Science and Software Engineering, Swinburne University of Technology, Australia

[3]Manchester Business School, The University of Manchester, United Kingdom

rickzh@gmail.com, john@cs.auckland.ac.nz, jgrundy@swin.edu.au {N.Mehandjiev, M.Carpenter}@mbs.ac.uk

Abstract—Supply chain modelling and simulation by SMEs (Small-to-Medium Enterprises) is a challenging problem. This is due both to complexity of the supply chain models required and the lack of required expertise among the SMEs. The problem is important since SMEs need to represent and modify their evolving skills and processes to be visible in electronic marketplaces and supply chain design platforms. We demonstrate how this problem can be addressed by developing a suite of novel domain-specific visual languages and a support tool. The challenging setup of our research context motivated us to trial a new approach for the design of our visual languages and to employ a collaborative development process across our distributed research team.

Keywords—domain-specific visual languages; visual language desige; supply chain modelling and simualation; visual language tools

I. Introduction

It is increasingly challenging for SMEs to compete successfully without well-designed and evolvable supply chain networks [1]. Many approaches aim to support the modelling of such networks. One example is the EC-funded project "SMEs Undertaking Design of Dynamic Ecosystem Networks" (SUDDEN). This provides an infrastructure for value-added process coordination and supply chain management for SMEs [1].

Due to the complexity of most supply chain models, non-specialist end users find them difficult to understand, build and reconfigure [2]. For example, the SUDDEN model has dozens of domain-specific concepts and relationships. A key challenge is to make these constructs and models based on them tractable for SME end users. One solution is to provide a suitable domain-specific visual modelling language (DSVL), extracting a simplified view of the underlying meta-model customised to the skills profile of our target end users; and developing an appropriate suite of support tools to assist them [3, 4].

Challenges for such a modelling platform include (i) a lack of generally accepted DSVL design and evaluation techniques to help ensure a "good" solution [5]; (ii) the research team for our project being highly distributed (UK and New Zealand, with our target end users in Austria); and (iii) the need to iteratively develop both the DSVLs and their support tools while gaining end user feedback. In this paper we describe a suite of DSVLs we have developed to support SUDDEN-based supply chain modelling and approaches used to develop this solution.

The key motivation of this research is to help SMEs to participate in complex collaborative supply chain network modelling. It is also to enable them to share their knowledge and obtain visibility of their skills and processes in online marketplaces and supply network design platforms. The SUDDEN domain model creates value via collaboration and coordination of SMEs participating in supply chain modelling and simulation [1].

Ye et al [6] present an ontology based supply chain management tool (Onto-SCM). They use IDEF5 for visual interface design, and Ontolingua to describe model semantics and knowledge communication and inter-connection support. Its main drawback is that IDEF5 has not been widely adopted since its introduction. Cope et al [7] propose a supply chain simulation solution tool, which provides generic GUI interfaces, supply chain simulation ontologies, and an automatic model generator. Its major drawback is that the tool uses XML schema to develop ontologies and the mapping between XML schema and semantic web can cause inconsistency.

II. Our Approach

We formed a collaborative research team comprising UK-based SUDDEN model and repository developers and New Zealand-based DSVL and modelling tool developers. We adopted a synthetic approach for our research project and tool development. This included using the Unified Process (UP), distributed SCRUM and XP. We adopted UP as a primary conceptual framework to aid us for analysis, design and evaluation. We used distributed SCRUM to manage and organize our project. Figure 1 shows our development process. From SUDDEN models (1) we developed a set of candidate visual languages using a set of principled approaches (2). We gained user

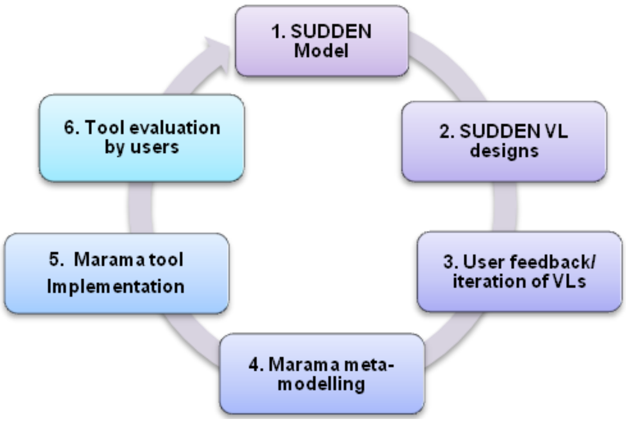

Figure 1. Our MaramaSUDDEN development approach.

feedback on candidate VLs (3) and then developed a set of meta-models in our Marama visual modelling meta-tool [8] to describe the VL elements and underlying SUDDEN constructs (4). We then developed a visual modelling tool using the Marama metatool, MaramaSUDDEN, including support for import and export of simulated supply chain specifications to a shared semantic web-based ontology (5). We evaluated the prototype tool with our target SME end users (6) and fed back results into both our visual language designs, design approach and underlying SUDDEN modelling concepts.

The key contributions of the research we describe in this paper include: 1) identification of a set of design principles for domain-specific visual languages enabling design of a range of SUDDEN DSVLs enabling evolution of extension of these DSVLs as the SUDDEN model continues to evolve; 2) rapid prototyping of a MaramaSUDDEN modelling tool enabling a continuous feedback approach on our evolving DSVL designs; and 3) use of a distributed, iterative process involving the SUDDEN research team, Marama research team and SUDDEN end users to develop DSVLs and support tool.

III. THE SIMULATION METHODOLOGY

We used an innovative approach to design a new DSVL for the domain model, including carefully chosen design rationales and paradigms [5]. Our chosen visual notation design paradigm includes a set of visual elements to express meanings. Our visual notation design paradigm includes primary notational elements **geometric shapes** and secondary elements (**colours** and **textures**), **lines** (various linear characteristics and variations), and **text** (providing labels for visual symbols). These elements have a range of values to provide design capacities for discrete visual symbols and collections of symbols which suits the problem domain.

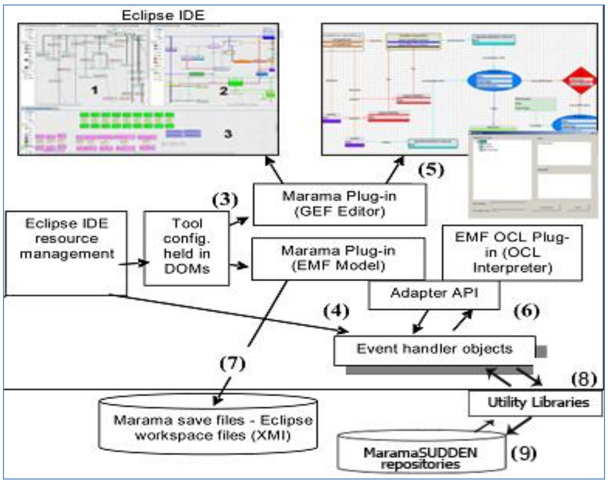

Figure 2. The architecture of the MaramaSUDDEN prototype

SMEs can use the MaramaSUDDEN DSVL to directly design, plan and manipulate their supply chains models. The results of the modelling activities will be saved into the semantic web based simulation ontologies for sharing and interaction among supply chain partners. The simulation ontology was developed in OWL format. It defines classic business activities involved in the supply chains, including product decompositions, resources assignment (materials and machineries), value creation, processes flow, supply chain forming and etc.

Figure 2 illustrates the four-layer architecture of the MaramaSUDDEN tool, including presentation, business logic, utilities and data. The presentation layer comprises a view based visual modelling interface, GUI windows, messages, browsers and etc (5), which is extended from the Marama meta-tools. To increase modelling scalability, we used a modularization/ abstraction technique to encapsulate shapes and views. The business layer (6) defines and describes the specific problem domain/sub-domains of the SUDDEN model by using the Marama tools. It also specifies the behaviours of the sub domains. The utilities layer (8) comprises different libraries and plug-ins we developed to control the simulation ontology in the data repositories. The support libraries are based on both Jena [9] and the Marama framework. The data layer (9) contains semantic web based ontologies. The ontologies are independent and neutral from any platform, therefore the repositories are portable and easy to share. To ensure users' data is consistent, a consistency management mechanism checks constraints against both the meta-tool data and the ontology repositories.

IV. THE MARAMASUDDEN DSVL

Evolution of the SUDDEN concept model meant that too many concepts needed to be represented for our initial single DSVL [3] to be practical. Instead, we split the one single complex view into three different view to allow users to solve different sub-domain issues, an application of Moody's Principle of Complexity Management [5]. The trade-offs of using this multi-view approach include an increased level of hidden dependencies, and consequently increased concerns over data consistency of instances when they are shared by different views. We feel that juxtaposition of views and other techniques for Cognitive Integration [5] can mitigate these deficiencies.

A. Notation design for the Product DSVL

We defined a Product DSVL to allow users to model complex product decompositions. Figure 3 shows a sample MaramaSUDDEN tool screen shot for the Product DSVL in use describing two alternative parts decompositions of a car door.

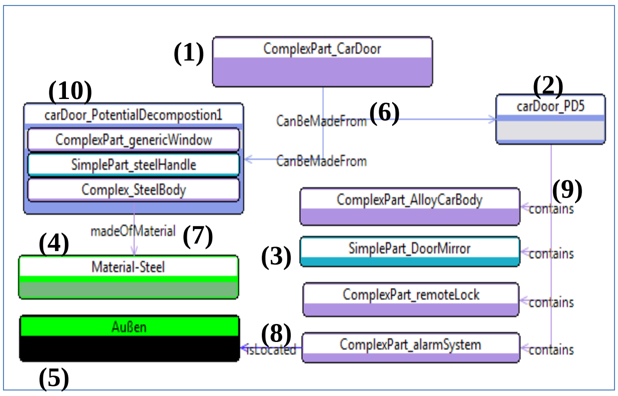

Figure 3. An example of a product DSVL diagram

In the SUDDEN domain model, “Product Decomposition”, “Complex Part” and “Simple Part” are children of concept “Value Creation” (hence they are coloured various hues of blue). We used rectangles and round rectangle with variations, e.g. thickness and different hues to distinguish elements.

In Figure 3, (1) is a “Complex Part” class, (2) is a “Potential Decompositions” class and (3) a “Simple Part” class. These are also child classes of “Value Creation” and share similar design concepts using colour variations to distinguish them. (4) is a “Material” class, a child class of “Resources”. Green is used as the colour theme for the “Resources” family. A symbolic texture pattern distinguishes it from (5), which is a “Location” class (another type of Resource: Aussen here is German for exterior). (6) is a “CanBeMade From” relationship between a “Complex Part” (1) and a “Potential_ Decomposition” (2) and comprises a labelled, coloured, arrow. The open arrow stands for the direction of the visual communication flow, and helps users to organise the diagram space. The text phrase is used to express the detailed semantics of the relationship. We used a bluish colour to indicate this relationship type connector is related to the “Complex Part” class and via that, the “Value Creation” family.

(7) is a “made of” relationship between “Materials” and “Complex Parts”. (8) is a “located relationship”, and (9) is a “contains” relationship. (10) is a modularized “Potential Decomposition”. This is an alternative and more compact representation of a decomposition over that of (2) and it contains relationships, which contains a set of nested “Complex Parts” and “Simple Parts”. This structure is another example of the use of Complexity Management [5]. We used vertical alignment layout conventions here to encapsulate classes to save screen space. The nested shapes within this “Complex Part” module can be identified by use of colour contrast techniques, layout/placement constraints and conventions to provide distinctions [3].

B. *Notation design for the Process DSVL*

The Process DSVL allows users to model and manipulate businesses processes of supply chain networks based on “Goals”. “Actors” are responsible for decision-making for these “Goals” [1]. For sub-notation design in this view we used three major colour themes with strong variations: green (for the “Resources” family); blue (for “Value Creation” family); and “Red” (for classes requiring additional precautions). We also used a range of textures, shapes and lines that provide clear distinction for each of the sub-notations. Figure 4 (left), shows a process view. A rectangular “Goal Decomposition” container (1) specifies two subgoals for managing a particular contract. An oval “Process” (2) describes how a particular subgoal can be operationally achieved. Both “Process” and “Goal” are child concepts of “Value Creation”, using the blue colour theme with variations of shapes and hues to provide clear distinction for child concepts in the same family. “Actors” such as (5) are given “Responsibilities” (13) over subgoals. Each subgoal requires a set of “Competencies” (9), which are matched to those declared (3) by Actors (close match indicating an Actor is suitable to be responsible for the subgoal). Actors make “Operationalisation Choices” (8, 11) over what process is used to achieve a subgoal (7). “Resources” (6) may be required to execute processes A “Machine” (10) is a particular type of “Resource”. (12) is a “fit” relationship, one of three basic process dependency types [10]. These dependencies are made distinctive using red or orange to alert users that these classes are critical to the process modelling. Relationships between different classes inherit the colour theme from the classes they originate from.

C. *Notation design for Abstract Supply Network DSVL*

The Abstract Supply Network (ASN) DSVL allows users to dynamically select partners, and form supply chain teams in a bottom-up approach. This lets suppliers form consortia that offer innovative combinations and bundling of activities. In Figure 4 (right) (1) and (3) are basic “Nodes” of virtual enterprise eco-systems [1]. (1) is a car lock supplier; (3) is a car door window maker. To form a complete supply chain team (4, 5), “Nodes” must recruit other supplier partners (2), in this case a steel car door body maker. (1) and (2) form a partial supply team (4); and (2) and (3) form another partial supply team (5) to potentially supply car doors. This information can be seen by all SMEs in the virtual enterprise eco-systems. Other suppliers (e.g. a car door handle maker) can offer to form a complete supply team to bid for a potential contract.

Team1 (5) and Team2 (4) form an ASN (6). (7) is a “Responsibility” class, e.g. On-time-delivery. Both “Nodes” and their “Partners” share “Responsibilities”.

V. Evaluation

We used a continuous evaluation approach [3] throughout the development lifecycle, to iteratively improve our DSVLs and tool, and to clarify the mode of tool usage by its end users. Initial feedback focused on simplifying individual views, reducing the number of

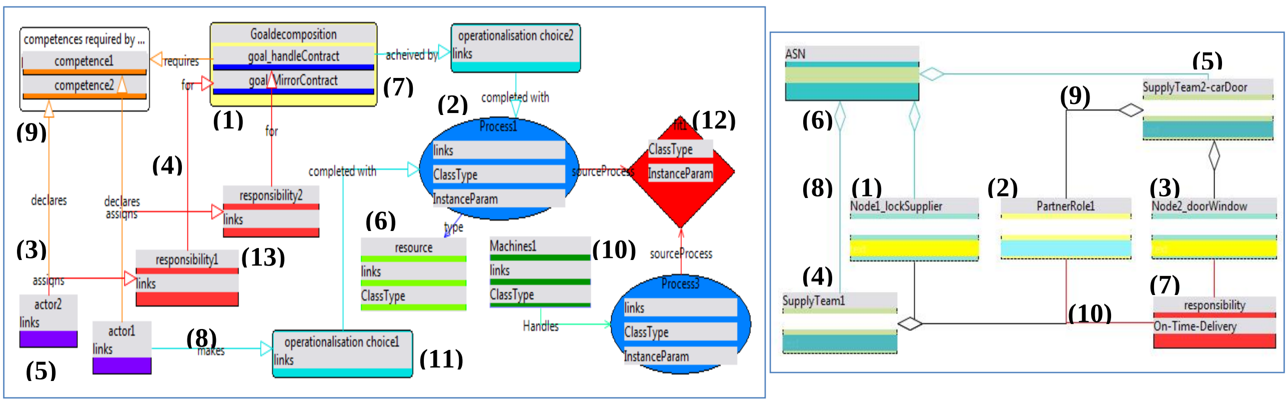

Figure 4. (left) example of a Process diagram, and (right) example of an ASN diagram.

palettes and replacing many specialised connections with a single “smart connector” which morphs into appropriate specialised connectors after linking shapes.

We evaluated the MaramaSUDDEN DSVLs and modelling tool with four SME end users in November 2009 in Steyr, Austria. The first stage of the evaluation concerned a number of questions after viewing a walk-through tour of MaramaSUDDEN. The aim was to solicit feedback on how important various potential evolution paths for the different parts of the knowledge base were and if the visual notation and interaction principles of MaramaSUDDEN were suitable for different end users.

Discussion took place during the walkthrough. One point of discussion was extending the scope of target end users to include specialist end users rather than just the more technically savvy platform owner. The right answer would probably depend on the balance between benefits and risks in allowing this kind of evolution. A moderated evolution process was proposed, where partners might evolve peripheral aspects of the ontology but these should be reviewed by the platform owner before committing into the central knowledge base. This identified additional workflow requirements that need to be supported in the next revision of the tool.

The support for the principle behind the tool was very strong; it was thought essential for platform owners to be able to manage the ontology as required. The suitability of the tool for direct use by general end users was more questionable. Since ontology editing is inherently complex, the tool was not specifically designed for use by all end users but rather by platform owners of SUDDEN. These end users are expected to be familiar with the knowledge structures driving the SUDDEN software, although at a relatively non-specialist level, for example they are not expected to make correct decisions about differences between ontology classes and instances, and between part-of and specialisation decompositions; as such these will be largely hidden from them. A simplified tree representation was identified as a representation which could be open to non-specialist end users. One such end user was present in this evaluation and thought the tool relatively easy to understand.

In contrast the ability of MaramaSUDDEN to also offer support for the editing of processes, especially those inside companies, was of more debatable use with the users. That was related to the business logic of doing so rather than a lack in the visual language facilities. A final question asked whether SUDDEN had achieved its original goal of producing a more flexible method for producing supply chains. The users agreed that SUDDEN had partially achieved this.

Finally some extensive testing of MaramaSUDDEN was performed. One of the target users, who fits the background and expected skills of a platform owner, was asked to perform a short scenario of modifying the ontology, and the group as a whole was asked to comment on these activities and their effects. The comments and interactions were videotaped for follow-up analysis. The results from this activity were broadly positive and overall the tool proved to be easy to understand and usable by the target user.

VI. DISCUSSION

The goal of our visual language development was to visualize complex domain-specific concepts and knowledge against supply chain modelling and share information among users effectively. Overall the notations of the MaramaSUDDEN DSVL appear to have accomplished this task. We separated the domain concepts into three distinct yet closely inter-related visual diagram types to aid the users in managing complexity. We adopted a distributed, iterative approach to engineering and evaluating our MaramaSUDDEN prototype tool. In addition, our desire to develop effective, scalable visual notations led to research on scientific visual notation and visual sentence design using the Physics of Notations framework [5]. This includes developing a set of consistent principles to underpin our visual language design; rapid prototyping and end user evaluation of our DSVLs.

Feedback we obtained from our target end users on our modelling languages and support tool was generally very positive. In future work we will conduct more formal usability evaluation. More tool features will be added for editing OWL repositories, improvement of syntax and semantic checks and ontology functionality.

REFERENCES

[1] N. D. Mehandjiev, I. D. Stalker, and M. R. Carpenter, "Recursive Construction and Evolution of Collaborative Business Processes," in *Business Process Management Workshops*, 2009, pp. 573-584.

[2] H. Min and G. Zhou, "Supply chain modeling: past, present and future," *Comput. Ind. Eng.*, vol. 43, pp. 231-249, 2002.

[3] J. Hosking, N. Mehandjiev, and J. Grundy, "A domain specific visual language for design and coordination of supply networks," in *Proceedings of the 2008 IEEE Symposium on Visual Languages and Human-Centric Computing*, 2008.

[4] L. Li, J. Hosking, and J. Grundy, "MaramaEML: An Integrated Multi-View Business Process Modelling Environment with Tree-Overlays, Zoomable Interfaces and Code Generation," in *Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering*, 2008.

[5] D. Moody, "The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering," *Software Engineering, IEEE Transactions on*, vol. 35, pp. 756-779, 2009.

[6] Y. Ye, D. Yang, Z. Jiang, and L. Tong, "Ontology-based semantic models for supply chain management," *The International Journal of Advanced Manufacturing Technology*, vol. 37, pp. 1250-1260, 2008.

[7] D. Cope, M. S. Fayez, M. Mollaghasemi, and A. Kaylani, "Supply chain simulation modeling made easy: an innovative approach," in *Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come*, Washington D.C., 2007.

[8] J. Grundy, J. Hosking, J. Huh, and K. N.-L. Li, "Marama: an eclipse meta-toolset for generating multi-view environments," in *Proceedings of the 30th international conference on Software engineering*, Leipzig, Germany, 2008.

[9] B. McBride, "Jena: a semantic Web toolkit," *Internet Computing, IEEE*, vol. 6, pp. 55-59, 2002.

[10] T. W. Malone, K. Crowston, and G. A. Herman, *Organizing Business Knowledge: The MIT Process Handbook*: MIT Press, 2003.