John Hosking, Nikolay Mehandjiev and John 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, Herrsching am Ammersee, Germany, 15-20 September 2008, IEEE CS Press

A Domain Specific Visual Language for Design and Coordination of Supply Networks

John Hosking[1], Nikolay Mehandjiev[3] and John Grundy[1,2]

[1]Department of Computer Science and [2]Department of Electrical and Computer Engineering
University of Auckland, New Zealand

[3]Manchester Business School, The University of Manchester, United Kingdom
{john, john-g}@cs.auckland.ac.nz, nikolay.mehandjiev@manchester.ac.uk

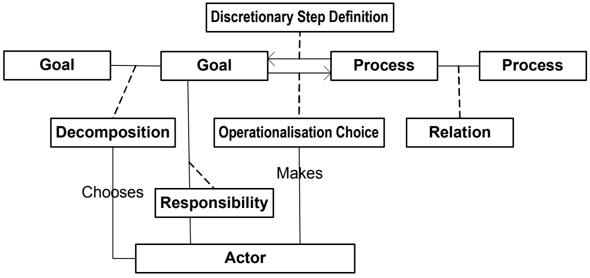
Abstract

We have developed a domain specific visual language (DSVL) and environment to support the modeling of small business-based dynamic supply networks. We describe our approach to the design of the DSVL, challenges faced, the implementation of a prototype environment, and preliminary evaluation.

1. Introduction

The planning and management of dynamic supply networks is a challenging problem. The EC-funded project SUDDEN aims to support SMEs to collaboratively assemble and optimize their supply networks [2]. In this project a UML-based domain model has been developed for supply network modeling and optimization [4]. To operationalize the model we have developed MaramaSUDDEN, a DSVL and modeling environment. We report here on the development of the DSVL and environment, focusing in particular on a variety of interesting notational design issues of relevance to similar types of DSVL.

The core of the previously developed SUDDEN process domain model is shown in Figure 1 (the full model is much more complex). Goals are decomposed into sub-goals by the responsible actors. A sub-goal is then characterized by the set of competences it requires, which are matched to the set of competences declared by another actor. If the matching proves positive, the other actor is given responsibility for that goal, and has to make a context-dependent choice on how to accomplish the goal. They may either choose from a set of alternative process models, devise a new process model or just formalize the inputs and outputs leaving the details unspecified (Discretionary Step Definition). The choice of “how” a goal is to be achieved is context-dependent, made and changed by the actor responsible for this goal, whilst process components (steps) remain linked to the goals they pursue to allow rollback. Some steps may have no other definition but the goal, thus opening up the scope for ad-hoc “discretionary” action. This separation and linking of goals, actors and process choices allows innovative bundling of services, where goods could, for example, be packaged “in transit” after passing electrical safety tests in distribution warehouses.



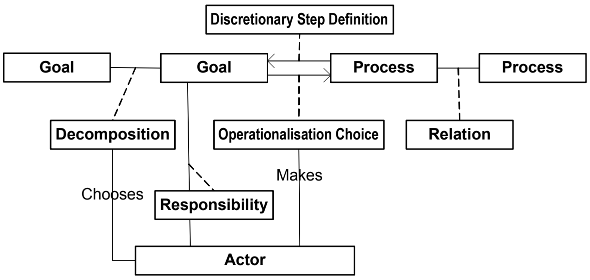


Figure 1: Core SUDDEN Domain Model

Moving from this theoretical framework to a practical modeling system, the SUDDEN team required a tool to support modeling of supply networks that conform to the SUDDEN domain model. This necessitated a custom surface modeling notation and a supporting modeling environment to allow modeling against the domain model. A number of peculiarities with the SUDDEN model complicate the development of a suitable surface notation. These include:

- the full domain model has a rich set of concepts;
- many concepts are relatively abstract, and so don’t immediately suggest a descriptive iconic form;
- there are many relationships between the concepts and careful notational design is needed to avoid crippling hidden dependency [3] problems;

- some concepts and relationships have existing semi-standard notations (e.g. for co-ordination) which need factoring into the notation design.

3. Our approach

Our approach to surface notational design is motivated by Cheng's Representational Epistemology (REEP) approach [1] which advocates:

- single visual representations with multiple cross-cutting content dimensions
- a strong task orientation to diagrams
- an emphasis on juxtaposition i.e. side-by-side presentation of information over creating hidden dependencies [3]
- a heavy use of orthogonal layout characteristics to exhibit and organise semantic meaning (raising layout from secondary to primary notation).

Our main reason for adopting this approach was the high number of cross-cutting relationships in the domain model. To apply the approach we evaluated each of the sub-notational elements considering how best to represent them orthogonally yet clearly. Different notational mechanisms available included:

- different icon and connector shapes and formats;
- layout constraints imposed on standard "icon and connector" notations (e.g. vertical or horizontal trees or horizontal or vertical "flow);
- containment, i.e. collections of sub-notation elements contained in other notational elements;
- ordering/structuring/layout of contained elements within an icon.

Each of the notational abstractions suits a different representational approach. Some relationships and concepts suggest well-established metaphors e.g. goal decompositions as cascading trees or process step co-ordinations as control and information flows. Others, such as operationalisation choices and their factors, have less obvious representations.

Our design method was an incremental iterative one, where at each iteration we: 1) chose a sub element for inclusion; 2) considered existing or new metaphors for the element; 3) considered how that metaphor could be adapted and orthogonally (visually) integrated with the existing combined notation (typically requiring adaptations to the existing notations to emphasise orthogonality and address new relationships); 4) evaluation through exemplar development. The iterations were each undertaken in collaboration with the SUDDEN domain model developers who provided advice on the acceptability of metaphors and the understandability of the integrations.

4. MaramaSUDDEN

We used our Eclipse-based Marama meta-toolset [4] at each iteration to prototype an environment for the SUDDEN notation. This allowed us to rapidly mock up substantial running examples and check that surface notation to model mappings were realizable. We began by initially modeling the full SUDDEN domain model using the Marama meta-model tool. Marama's visual element and view tools were then used to: develop new notational elements (icons and connectors) or modify existing ones; incorporate them into a view editor; and map them to the meta-model. Element layout was initially unconstrained so different arrangements (e.g. directional flow, or element containment) could be experimented with. Marama constraints were then applied to implement some layout approaches (e.g. enforcing containment or ordering). In other cases layout approaches were adopted as unenforced *conventions* recommended to the end user modeller. The latter was in situations where the layout constraint was difficult to express as a Marama constraint or some user discretion was deemed to be valuable.

Figure 2 is a screen dump from the final iteration of MaramaSUDDEN. The model represents one approach to satisfying the high level goal of an SME to produce and sell a new all terrain vehicle. The icons at the left show a decomposition of that initial goal into subgoals related to marketing and sales, engineering and so on (1). These subgoals require various sets of competencies for fulfillment, such as a set of business analysis competencies (3) needed for the marketing and sales planning related goals. The decomposed subgoals lead to operationalisation choices such as a procedure to establish performance metrics (5). Various factors (6) may have affected this choice and the procedure is realised in the form of a set of coordinated process steps (7) which create, consume or share resources such as sets of requirements or metrics. The process steps may coordinate with other processes (8).

Several orthogonal sub-notations are evident, emphasized by icon form, color, layout and placement:

- Goals, Decompositions (GDs) and Competences: an ordered list of goal icons is contained in a GD icon (constraint enforced) (1) connected to a goal (2). A left-right layer and top-bottom ordering *convention* applies to the decomposition hierarchy. A convention allows some placement discretion (e.g. the initial goal decomposition). Competence sets (3) contain individual competence icons (similar to GDs) and are linked by connectors to goals requiring those sets. Competence sets may be elided (4). This introduces hidden dependencies but the benefit of enhanced screen real estate outweighs the hidden dependencies created.
- Operationalisation Choices (OCs) and Factors:

each OC is represented by an icon, connected to the goal it is "implementing" (5). A left-right convention emphasizes this is an extension of goal decomposition into the operation space. Factors are represented as a Factor set contained in and placed at top left of the OC icon (6) (constraint enforced). Individual factors are contained as an ordered list in the Factor set (constraint enforced). Factor sets can also be elided.

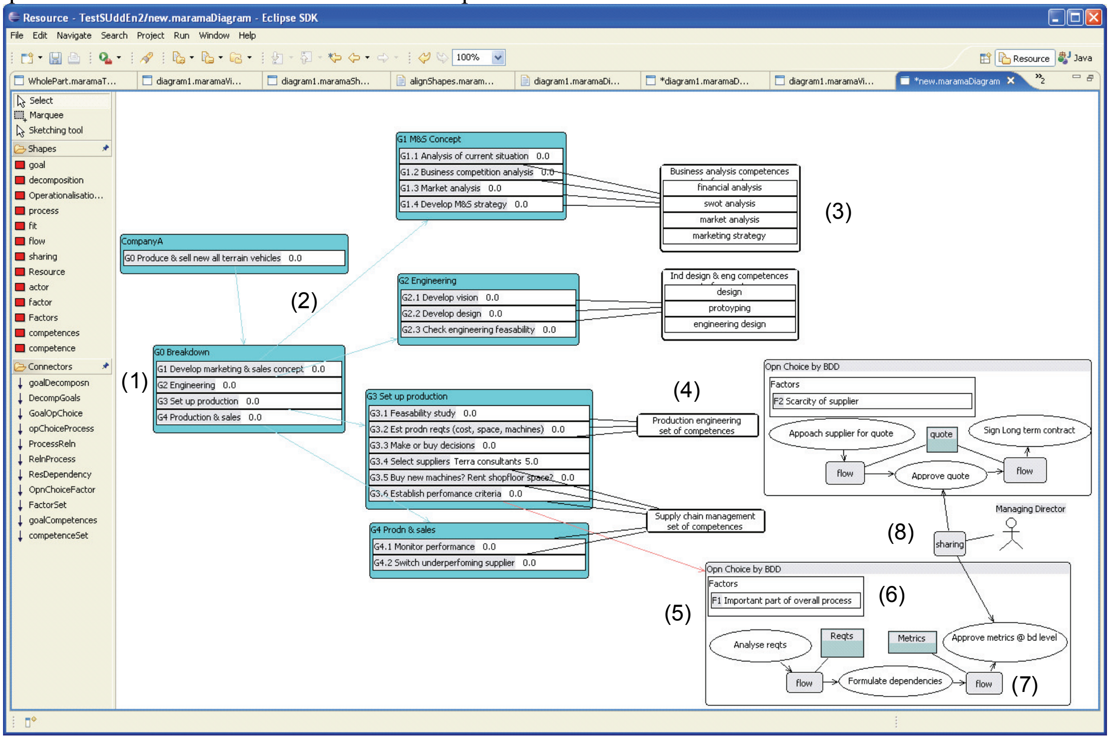


Figure 2: MaramaSUDDEN and simple supply network coordination model

- Processes and their Co-ordinations: Processes are associated with OCs, but their steps are coordinated by flow and sharing relationships that may span different OCs and may have associated Resources, including Actors. The approach taken is to enclose process steps within OCs (7) (constraint enforced) but allow dependencies and dependency relationships to be either contained or span between (8) OCs. A left right convention emphasizes fit and flow dependencies, with a superimposed vertical layout convention for sharing dependencies. OC elisions retain process dependency transitions: these exit the OC reflecting that an elided process step is the origin of this transition.
- Actors and their expressed Competences and allocated Responsibilities: Actors are represented by a UML-style Actor icon and are associated with process dependencies (8) and also with Goals they are responsible for (not illustrated). We have chosen not to represent actor competences and resulting responsibilities directly in this diagram. These are inherently associated with the process for matching actors to goals. A tabular metaphor for managing this has already been proposed [2] which we felt superior to an iconic/graphical approach. The hidden dependencies resulting from this violation of the REEP methodology are minimal compared to the efficiency gain from an additional purpose developed representational view.
- Resources (including Actors) and their Dependencies: Resources are associated by a connector link with the dependency relationships between process steps (7). Actors are a special type of resource and hence can also be associated in this way (8).

Several other considerations are apparent. First, for scalability, only a minimal attribute set is visible for each element: an evocative name, and important data needed for understanding (e.g. the matching value for goals). Other attributes are accessible via property sheets. Again, we made a considered trade-off between diagram scalability versus hidden dependency impact. Secondly, elision is important for scalability. The domain model complexity places heavy demands on screen real estate hence hidden dependency costs are outweighed by scalability benefits. Thirdly, there is low variation in icon shape. Instead we have relied on layout/placement constraints and conventions to provide distinctions. Despite the similarity in iconic shapes elements are readily identifiable and little effort needed to visually "index/search" the diagram. Finally, we found *conventions* important. We could enforce vertical alignment for the levels in the GD hierarchies but this would produce less readable diagrams.

5. Evaluation

We used a continuous evaluation process at each iteration step. One primary end user was involved heavily in the iterative design, giving feedback on notation choices and usability, and modeling examples to assess representational adequacy. In addition we have been guided by the Cognitive Dimensions framework [3], consciously choosing dimensions to emphasize and the resulting trade-offs and mitigations needed. We have emphasized the minimization of the *hidden dependencies* dimension in our design as inspired by the REEP approach. This combination of continual evaluation approaches, end user centered and theory based has given us confidence that our design process has led to an effective, usable modeling environment for the SUDDEN project. A more extensive evaluation is planned.

6. Discussion and Conclusions

This project had the aim of developing a notation and proof of concept implementation for modeling against the SUDDEN meta-model. The development of MaramaSUDDEN has achieved that aim. In the process it has given additional insight into notational design.

We adopted the REEP approach [1] (complex single notation) over past advocacy of multiple views [5]. We feel the complex cross cutting relationship and concept structure of the domain has justified this approach. We did not adopt the approach completely: the use of elision and a separate tabular representation for the competence matching process violate REEP philosophy. We feel these violations were justified by ensuing efficiency and viscosity reduction gains. In addition, we expect SUDDEN modelers to use multiple views to express *alternative* decompositions as we feel that expressing such alternatives on the one diagram would be confusing for end users. Mitigation of the hidden dependencies resulting from common partial decompositions could be via juxtaposition of views and/or visualizations of diagram differences [8].

A primary factor in the success of our notational approach, however, is the elevation of layout to first class status. This has significant implications for meta-tools such as our own Marama toolset. While most such toolsets have good support for conventional internal iconic layout, support for more global layout/placement primitives is typically limited. Minas' Diagen [9] and our own Kaitiaki [4] works are some of the few approaches that focus on this problem. However, they are relatively low level in their orientation. We feel that provision of more appropriate layout primitives is a significant research imperative for meta-tool researchers. In advocating this, we should emphasize that there is no shortage of available layout algorithms. The real issue is a need for end user oriented metaphors that make layout primitives and their combination accessible to meta-tool users. Layout By Example is one of very few techniques that approach the level of abstraction we feel is needed [10].

In current work, we are exploring a range of these issues and have been experimenting with other applications where the REEP approach is applicable, including an environment for EML, a business process mapping notation we have been developing [6].

The continuing work on the SUDDEN system is now focusing on providing automated support for network design and optimization using the models created by MaramaSUDDEN, and on the interactive mode of using the MaramaSUDDEN modeling facilities when run-time changes in the model and hence in the system are to be implemented, for example changing one supplier or coordination process with another.

References

[1] Barone, R., & Cheng, P. C.-H. (2004). Representations for problem solving: on the benefits of integrated structure. Proc 8th Intnl Conf on Information Visualisation (pp. 575-580). Los Alamitos, CA: IEEE.

[2] Carpenter, M., Stalker, I., Mehandjiev, N., de Miguel, V.B., Kouvas, G., Peherstorfer, P., Weichart, G., Coordination Models and User Interaction Language, Deliverable D2.1 SUDDEN project, April 2007.

[3] Green, T.R.G. & Petre, M. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. JVLC 1996 (7), pp. 131-174

[4] Grundy, J.C., Hosking, J.G., Liu, K., Huh, J., Marama: an Eclipse meta-toolset for generating multi-view environments, Proc ICSE08, Leipzig, 2008, 819-822.

[5] Grundy, J.C., Hosking, J.G., and Mugridge, W.B., Inconsistency Management for Multiple-View Software Development Environments, IEEE TSE, 24(11), November 1998, 960-981

[6] Li, L., Hosking, J.G., Grundy, J.C. 'Visual Modelling of Complex Business Processes with Trees, Overlays and Distortion-Based Displays', Proc VLHCC'07 Idaho, 2007, p.137-144

[7] Malone, T.W. and K. Crowston, The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, 26(1)

[8] Mehra, A., Grundy, J.C., Hosking, J.G. 'A generic approach to supporting diagram differencing and merging for collaborative design', Proc ASE'05, Long Beach, 2005, p 204-13.

[9] M. Minas and G. Viehstaedt, "Specification of Diagram Editors Providing Layout Adjustment with Minimal Change," Proc. VL'93, 1993, pp. 324-329

[10] K. Sugihara et al., "Layout-by-example: a fuzzy visual language for specifying stereotypes of diagram layout," Proc VL'92, 1992 pp. 88-94.