Models for visualisation of Complex Information Systems

David Db Bihanic, Polacsek Tp Thomas

Models for visualisation of Complex Information Systems

David Bihanic
CALHISTE Laboratory
University of Valenciennes and Hainaut-Cambresis
Valenciennes, France
david.bihanic@univ-valenciennes.fr

Thomas Polacsek
Information Processing and Modeling Department
ONERA – The French Aerospace Lab
Toulouse, France
thomas.polacsek@onera.fr

***Abstract*—Modular approaches (objects, components and service-based) offer efficient solutions to face the complexity of Information Systems. However, today they involve major problems in dealing with data/information and decision-making (from the standpoint of the end- user). In this article, we aim to initiate a dialogue between digital/interaction design and model-driven engineering through the study of a new data representation-visualisation approach based on generative visualisation. Due to recent advances in graphical user interfaces and interaction techniques, we expect to prove that the world of modeling could benefit from creating views of existing models defined in a modeling language complying with Meta- Object Facility (like UML).**

Keywords-component; Complex information system, model information, data representation

I. Introduction

Today, Information Systems (IS) are mostly composed of heterogeneous and interrelated subsystems including interactions (which evolve over time) between system components and external environment. Consequently, understanding data relationships and visualizing, perceiving the intrinsic logic of systems has become more complex. Therefore, the development of specific views and viewpoints adapted to the context is now crucial in order to guarantee optimal use of complex systems.

In this article, we will show, on one hand, that we could consider current IS as real complex systems. We will then present the premises of a study aiming to define a new approach of visual graphical representation-visualisation of large-scale information systems (which are inevitably complex systems).

II. Complex systems (definition)

Given the plurality of complex systems, there is no single definition valid for all systems. The national network of complex systems defines it as: "any system comprised of a great number of heterogeneous entities, among which local interactions create multiple levels of collective structure and organization."[1] We can therefore consider a complex system as a system that features some of the following properties:

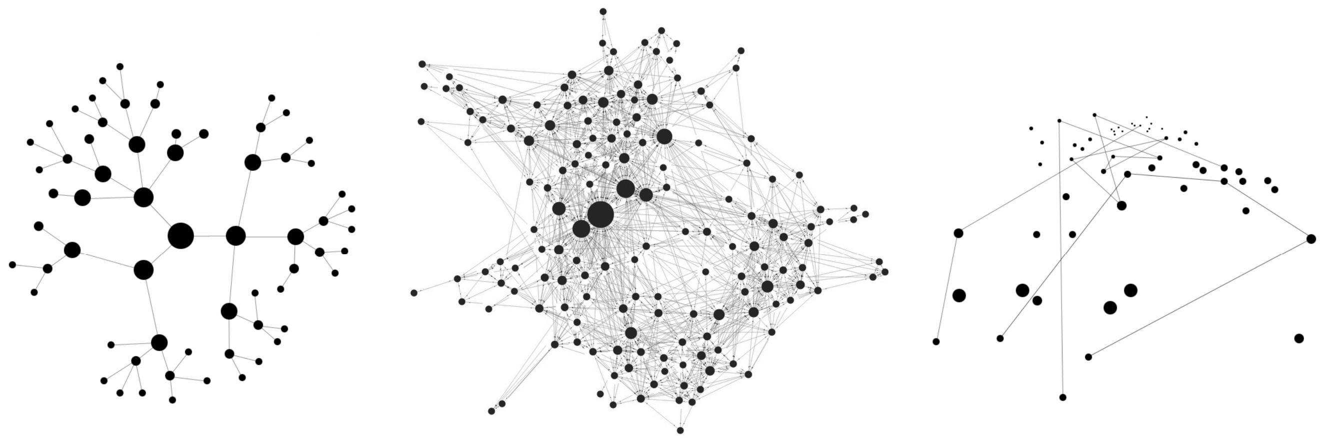
- *Heterogeneousness*, a complex system is composed of several entities or agents. These entities generally belong to different types and have a particular internal structure,
- *Flow processing*, the various elements of the system having relations or interactions between them,
- *Size*, (although this is not a fundamental characteristic) the size of a system and therefore its associated processings may be a feature of complexity,
- *Hierarchical organisation*, complex systems present hierarchies that form networks consisting of interrelated elements or agents and they interact,
- *Evolution*, complex systems have feedback, collective behaviour and emergent properties.

In a complex system, the properties and behaviour of its individual elements are not sufficient to predict its overall behaviour. Any complex system, whether it is an ecosystem of living organisms, the organisation of an urban community or a network of actors, is defined through its characteristics and its environmental properties (in other words, through its relationships between data and its interactions between elements).

Today, data is big, and it's all about "connecting the dots." In the past few years, big data has essentially gone from zero to hero in the enterprise tech world. Except for one thing: in real terms, it hasn't. Many seem to have forgotten that big data was around, and being put to good use. Unquestionably, enterprise data volumes have grown immensely, and organisations have indeed begun to recognize the value hidden in these larger stores. According to a 2011 study by the Aberdeen Group, organizations that effectively integrate complex data are able to use up to 50 percent larger data sets for business intelligence and analytics, to integrate external unstructured data into business processes twice as successfully, and to slash error incidences almost in half. The connection between a company's success and its ability to leverage big data is very clear.

One still has a long way to go. Information Systems (IS) and their models have become too complex to be simplified and understandable by humans [5]. Many works have focused on the notion of view and viewpoint, either in information systems development, knowledge representation and software engineering. If we refer to the

[1] The French National Network of Complex Systems, n.d., <http://rnsc.fr> (accessed May 5, 2012)



(a) (b) (c)

Figure 1. Three major families of representation *(a)* trees *(b)* maps *(c)* landscapes

IEEE standard 1471, a view is "a representation of a whole system from the perspective of a related set of concerns" and a viewpoint "a pattern or template from which to develop individual views by establishing the purposes and audience for a view and the techniques for its creation and analysis." One will note that the notions of view and viewpoint are not new [8], the idea is that an object is perceived from a viewpoint, which is that of the observer. With the IS becoming increasingly complex, the notion of viewpoint may help manage and use the complex systems [15]. Let us note that a technique of viewpoint creation should not only allow for a change in granularity but also for a shift in semantic universe.

III. Complex Data Visualisation

A. *Visual representation of complexity*

For over thirty years, many laboratories in computing sciences, specialised in human-system/machine (HCI), engaged into research programs in order to find new information and semantic knowledge representation models, as well as innovative data visual description solutions. Thus, several studies proposed original data visualisation methods, processes and techniques entirely dedicated to scientific statistics, metrics and analytics. In those days, data visualisation was considered as a specific tool (for scientific use only), a means of handling raw data sets and perceiving phenomena (data are evidence for phenomena).

Fifteen years ago a cross-disciplinary research initiative appeared. Its aim was to define a new visual/graphical information representation, theories and paradigms joining multiple experts in computer sciences, informatics engineering and also cognitive sciences, experimental cognitive psychology, etc. According to [2], "information visualisation (collectively called InfoVis) is the use of computer-supported, interactive, visual representations of abstract data to amplify cognition."

InfoVis first aimed at delivering cognitive processing advantages. The emphasis was on the process, not on the product. More than a tool, InfoVis was viewed as a way to "make sense", to reveal the meaning of data, and as well as its semantic similarities/relationships, this through formal representation and mapping. In other words, this information visualisation revolution proposed to transform semantic networks obtained from multiple data sources into visually perceived forms, in order to achieve a new and understandable vision of data, both practical and useful.

B. *Information visualisation: trees, maps and landscapes*

Through user's perception, information visualisation "amplifies cognition" by expressing semantic connections between data. It is based on techniques linking the model and its visual/graphical representation. As shown in the excellent paper by [6] (offering a new taxonomy of tools that support the fluent and flexible use of visualisations today), the mode of reasoning and of treatment offered to the user rely on these representation techniques, through possible views and viewpoints about the data. Each technique shows the complexity of both structural and semantic data in a specific mode. Each one defines the rules of organisation and visualisation of data and implements multilevel exploration processes combining multiple visualisation modes, and/which are reflecting a relationship between the data organisation, the semiotic realities and the conditions of appropriation by the user.

There are three major families of representation: *(a)* trees, *(b)* maps, *(c)* landscapes. Trees provide a hierarchical representation of data (heritage trees). Maps suit reticular representation (i.e. network). Landscapes provide multiscale representation of planar shapes (both contextual, logical, i.e. data flow diagram, and anthropocentric).

Tree representations of hierarchical data are used to organize, schedule or classify data. They rank among the oldest representations [17]. They were used very early in taxonomies (refer for instance to the tree of life of Aristotle in Historia animalium) and are still in common use.

Map representations (or reticular representations) with graph and multigraph indicate a strong connection

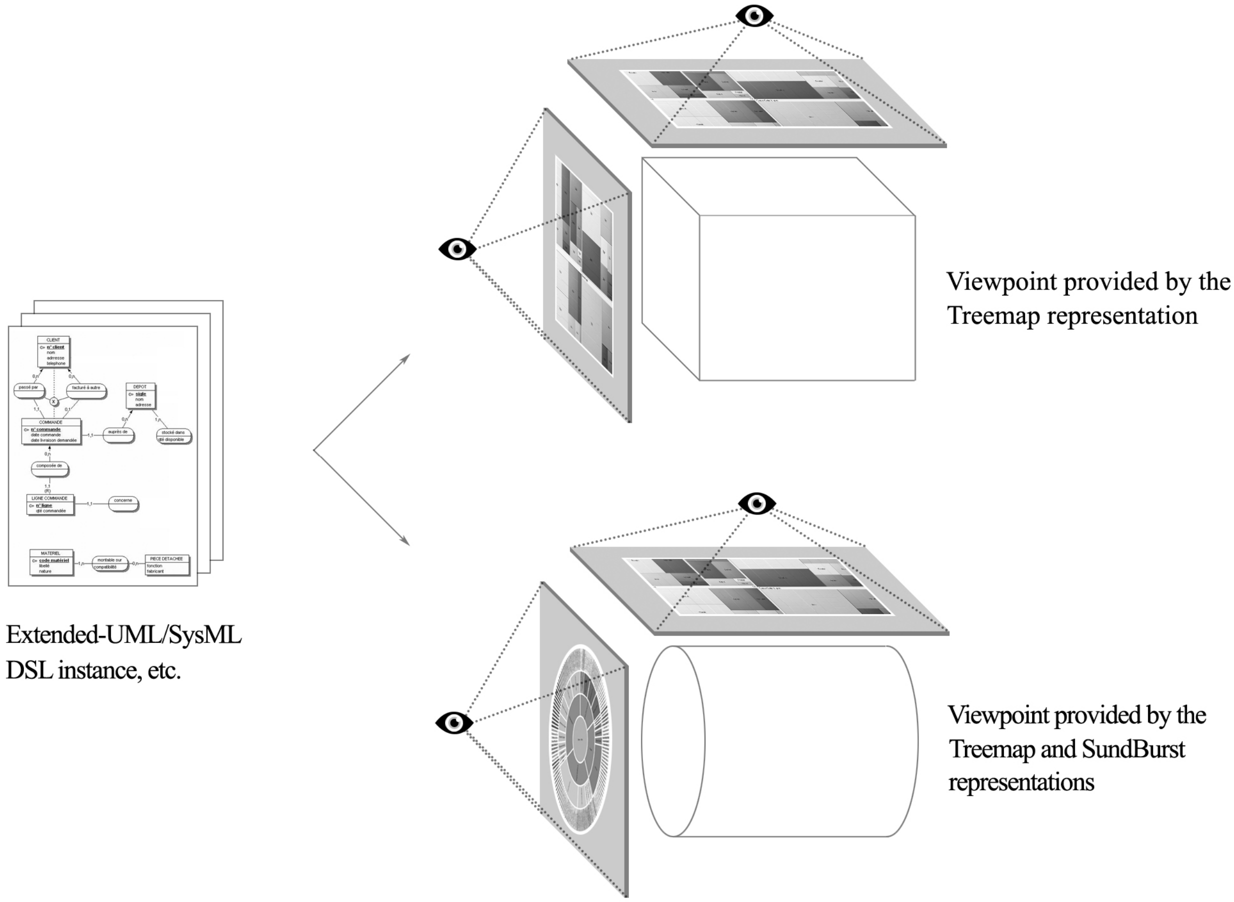


Figure 2. IS: transformation from model to visualisation

between data. An obvious example is the increasingly dense network of airline routes. Reticular representation affords the visualisation of strongly interconnected places, interdependent geographical areas, as well as the logical organisation of the network.

Data landscape representations are used to show space, time and/or semantic logical relationships through the display of data side by side. The landscape of data can be used, for instance, to visualize contextual relationships between data or semantic similarities.

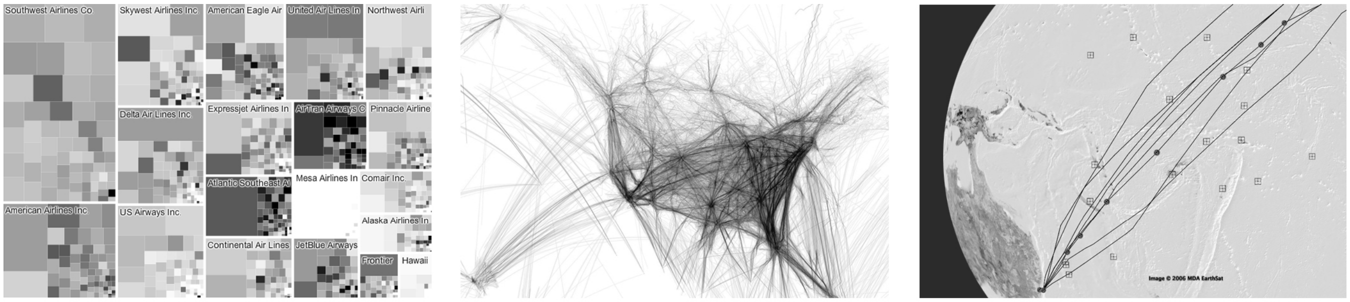
Each of these three families includes variants and adaptations. For instance: Treemaps [16], Cone-Trees and Cam-Trees [14], Hyperbolic Trees [7] [10], from DataScape to DataMountains [13] [4] etc. All these variants may come in different dimensions (1D, 2D, 3D, 4D), different interaction modalities (navigation, linking, brushing, etc.) and user tasks (filtering, retrieval, sorting, etc.)

IV. COMPLEX IS: CUSTOMISED VISUALISATION

As previously shown, many possible visualisations of complex data exist. Our aim is to provide a customized visualisation for a given user by giving him quick access to the relevant data, in an IS dedicated to him and this for each task. For this, we must define generic operators in order to restructure the visualisation of the IS in line with the user's needs. The first step is to define the predefined representation of the IS which constitutes for him a "pertinent" data semantic representation (such as hierarchical/tree, graph/map or landscape). After creating this view, the second step is to combine the view and ISs data dynamically to create contextual views based on the user's context. These contextual views take into account user's parameters such as task, localisation, visualisation equipment, etc.

A. *From model to predefined representation (…)*

It is now possible to create automatically predefined representation from models of the IS. For this purpose, one needs semantic information for representation. This information must be added to the IS model and this addition must be an extension of existing model notations. The aim is to evolve from the IS model to the best suited graphical representation meeting the needs of the user. To this effect, a graphical programming needs to be defined [11] [3] [18], in order to associate each relevant data of the IS model with a graphical representation (with a position, a size, an area, a



(d) Airline Treemap

Airline and airport traffic and delays
Creds: JuiceKit™

(e) Air traffic network

The paths of air traffic (US) visualized in color and form
Creds: A. Koblin

(f) Flight tracker data landscape

Tracking maps/status and airport delays for airline
Creds: S. D. Jackman

Figure 3. Examples of viewpoints (airline IS) - *(a: d)* trees *(b: c)* maps *(c: d)* landscapes

radiance, etc.) and to associate each relevant relationship between data in the IS model with a graphical representation (hierarchical, reticular, etc.)

In an airline's IS, all information is not intended for the same business use and quite obviously no user will ever need all the information. For each user, one needs to design a user visualization entirely dedicated to his task, with an efficient visual representation. For instance, to analyse the seasonal market effect, a user needs a view over passenger traffic, airline routes, benefits and costs within a one-shot representation. Conversely, for maintenance tasks, there is no need for commercial information, but for more information on the state of the plane. The Line Maintenance Engineer will need an overview on the status of a specific aircraft, the time for maintenance, the list of tasks to perform, the explicit references, etc. The Maintenance Expert needs a global view of the entire airline fleet with location and state information and information on spare parts available to provide routing plans in case of failures.

Significant advances have been made in the tools of transformation models, with tools such as Kermeta[2] [9] or ATLAS Transformation Language (ALT)[3] [19], and the transformation languages with the Query View Transformation (QVT) standard [12]. It is possible to define generic transformations to transform the model that represents the IS in a graphical representation. Note that these transformation rules need to be defined in a higher-level, which is the modeling language level. From models expressed in conventional modeling languages like UML, it is possible to provide a coherent contextual graphical representation of the system for the user. Defining meta-languages to design graphical representations and their related transformations still remain.

B. *(…) and from predefined representation to visualisation*

As we have seen, there are several possible representations for information. Representations have many advantages, but are unsuitable for complex data visualisation and users' needs. One must define generic models, for data graphical representation, that allow a restructuring of the visualisation based on users' needs. For this, one needs to integrate some/enough semantic information in the IS model to allow, in the first instance, the automatic building up of IS data visualisation (with tree, landscape, etc.), and then in a second step, the definition of the true representation, given to the user, depending on user profiles incorporating different parameters such as activity, task of the user, his location, accreditation, etc. and hardware settings, such as the size of his screen. If the user uses a touch pad, a laptop or large screen, he operates in mobility or not, cooperative/collaborative or not, the GUI will change for a better usability and, consequently, a higher performance. The concept of "pertinent" visualisation is a visual-graphic adaptation of the complex semantic-representational for a specific user in a specific context.

In the example of an airline IS and its maintenance tasks, the Line Maintenance Engineer is in a mobility situation and has a limited time allotted for his tasks. More than a simple view, we need here a visualisation dedicated to the size of the screen of the terminal, interaction modalities with the terminal, the information required (contextual information), etc. For the Maintenance Expert (with a global view), the visualisation displays more data with locating information, taking into account the larger screen of the user.

Thus, data should no longer be visualized by predefined representations, but with dynamic representations. Dynamic representations provide views, dynamically generated, with the appropriate data in the appropriate context. Therefore, one should move from predefined representations toward a dynamic data representation that would dynamically generate dedicated contextual views, sensitive to the different contexts and situations met. To be able to manage and master the IS complexity, one must centralise the processes within a contextual interface automatically created from the IS model.

[2] Kermeta (Triskell team), n.d., <http://kermeta.org/> (accessed May 5, 2012)

[3] Eclipse (Eclipse Foundation), n.d., <http://www.eclipse.org/atl/> (accessed May 5, 2012)

V. A VISUAL UNIFIED MODELING LANGUAGE (VSML)

Our purpose is to define the switch from an IS model to a representation-visualisation model [figure 2] – the last model is based on specific visual/graphical codifications (semantically-coherent and adapted to multiple user profiles) called "VSML" (Visual Semantic unified Modeling Language) – To reach this transformation, we introduce new semantical concepts (fully preserving all semantic correlations) providing visual and graphical rules. It refers here to an original graphical rule-based programming language based on event-driven graphics (without a fixed formal semantics) rewriting that is suitable for expressing user interaction. Thus, these rules allow to support transition from a source model writing in a specific modeling language to a corresponding graphical-based representation-visualisation language (modifying rules by other rules as well as graphical objects). It comes as a result of a careful study dedicated to InfoVis systems and techniques since the early works of [1]. Thanks to this survey, we detect several typological patterns and figures that lead to design new graphical representation/visualization models.

To automatically generate a graphical representation of a system, we need an IS model referring to the data organisation and also a user to identify relevant entities with "semantic tags." These tags allow us to define generic rules in order to perform the model transformation expressed in VSML. Here, there are two main inputs: the semantic data structure and the user profile (which means the definition of end-user device, user preferences and usage environments).

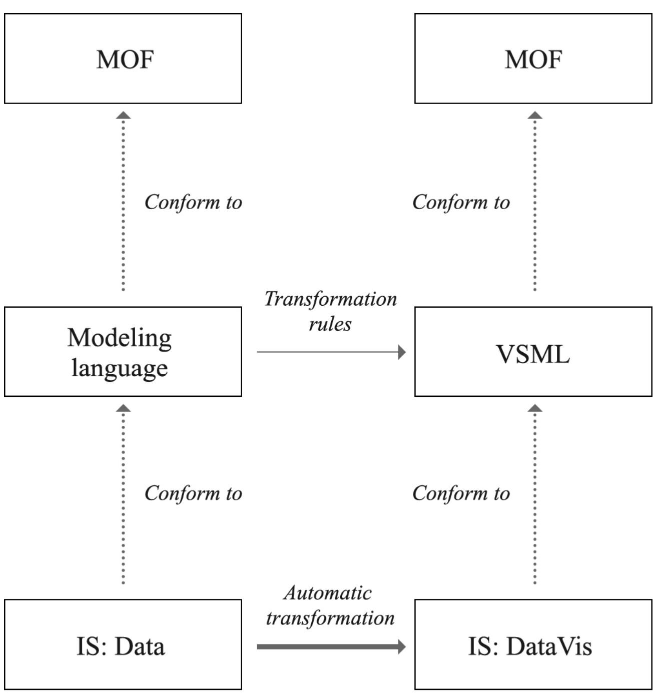


Figure 4. VSML transformation

Consider, for instance, a system with a lot of data which multiple relationships (like composition and aggregation). The user wants to see only hierarchical relations and he has a large screen display. The transformation should therefore give, for instance, a "Treemap" visualisation, where composition and aggregation should appear as a set of nested rectangles (adjacency relations in hierarchical data). In another case, if the user has only access to a mobile device such as mobile phone with a small and low-resolution screen display, the VSML transformation should give a "Sunburst" visualization offering the user to focus his attention on a sample of relationships between data entities.

Lastly, when the different model-to-model transformations are executed (making use of the most recent advances of Model Driven Engineering), the "conversion" of selected data into graphical assets is realized automatically (turning organization data model into readily understandable graphical representations).

VI. CONCLUSION

In this article, we have initiated a dialogue between the world of computer design and Model-driven engineering. Due to recent advances in user interface, we explained how the world of modeling could benefit from the creation of views of existing models. We tried to show that, for complex data, predefined views are not appropriate, and that semantic information needs to be added to existing models to support the creation of contextual views.

ACKNOWLEDGMENT

This work was supported in part by the French National Network of Complex Systems (RNSC) and the French Institute of Complex Systems of Ile-de-France (ISCPIF).

REFERENCES

[1] J. Bertin. *Semiology of graphics*. University of Wisconsin Press, Madison, 1983.

[2] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc., London, 1999.

[3] Ed H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS'00, pages 69, Washington, DC, USA, 2000. IEEE Computer Society.

[4] M. P. Czerwinski, M. van Dantzich, G. Robertson, and H. Hoffman. The contribution of thumbnail image, mouse-over text and spatial location memory to web page retrieval in 3d. In *Proceedings of Interact'99*, pages 163–170. Springer, 1999.

[5] C. Gershenson, D. Aerts, and B. Edmonds. *Worldviews, science and us: philosophy and complexity*. World Scientific, Washington (EU), 2005.

[6] J. Heer and B. Shneiderman. Interactive dynamics for visual analysis. a taxonomy of tools that support the fluent and flexible use of visualizations. *Queue - Micoprocessors*, 10(2), february 2012.

[7] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI'95, pages 401–408, New-York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[8] M. Minsky. A framework for representing knowledge. In P. Winston, editor, *The Psycology of Computer Vision*. McGraw-Hill, New York, 1975.

[9] P.-A. Muller, F. Fleurey, D. Vojtisek, Z. Drey, D. Pollet, F. Fondement, P. Studer, and J.-M. Jézéquel. On Executable Meta-Languages applied to Model Transformations. In *Model Transformations In Practice Workshop*, Montego Bay Jamaica, 10 2005.

[10] T. Munzner, E. Hoffman, K. Claffy, and B. Fenner. Visualizing the global topology of the mbone. In *Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS'96)*, INFOVIS'96, pages 85, Washington, DC, USA, 1996. IEEE Computer Society.

[11] C. North, N. Conklin, K. Indukuri, and V. Saini. Visualization schemas and a web-based architecture for custom multiple-view visualization of multiple-table databases. In *Information Visualization, Palgrave-Macmillan*, pages 211–228, 2002.

[12] OMG. *MOF QVT Final Adopted Specification*. Object Modeling Group, 2005.

[13] G. Robertson, M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, and M. van Dantzich. Data mountain: using spatial memory for document management. In *Proceedings of the 11th annual ACM symposium on User interface soft- ware and technology*, UIST'98, pages 153–162, New York, NY, USA, 1998. ACM.

[14] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, CHI '91, pages 189–194, New York, NY, USA, 1991. ACM.

[15] M. Shaw and D. Garlan. *Software architecture: perspectives on an emerging discipline*. Prentice- Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[16] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11:92–99, January 1992.

[17] J. J. van Wijk, F. van Ham, and H. van de Wetering. Rendering hierarchical data. *Commun. ACM*, 46:257–263, September 2003.

[18] F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. ManyEyes: a Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, Novem- ber 2007.

[19] D. Wagelaar. Composition techniques for rule- based model transformation languages. In *Proceedings of the 1st international conference on Theory and Practice of Model Transformations*, ICMT'08, pages 152–167, Berlin, Heidelberg, 2008. Springer-Verlag.