Evaluation of Multimedia Stream Processing Modeling Language from the Perspective of Cognitive Dimensions

Anna Bobkowska
Department of Software Engineering
Gdansk University of Technology
annab@eti.pg.gda.pl

Michał Nykiel
Department of Computer Architecture
Gdansk University of Technology
michal.nykiel@eti.pg.gda.pl

Jerzy Proficz
Academic Computer Centre
Gdansk University of Technology
j.proficz@task.gda.pl

Keywords: POP-III.C. cognitive dimensions, POP-V.B. case studies

Abstract

This paper reports on application of cognitive dimensions to evaluation of a visual modeling language for multimedia stream processing, called MSP-ML, during the construction of the language. It includes presentation of the context of evaluation in the VML engineering process and motivation for MSP-ML, the method of evaluation, the results of evaluation and comparison of the results with results achieved with other methods using VML quality criteria.

1. Introduction

During the construction of a new visual modeling language (VML), VML designers have freedom to define elements of the language. When defining the language, they may consider several perspectives of VML quality including cognitive fit of the VML under construction to its users. This perspective is very important as it has impact on efficiency of modeling as well as satisfaction of the users. But on the other hand, it is very difficult to capture. Several terms are in use for description of the concepts from this conceptual space, e.g. usability, cognitive fit, natural mental representations or intuitiveness for users. A set of cognitive dimensions (CD)[4] is a tool for examination of notations against their cognitive aspects. The questions are: Are CDs useful for VML designers for constructing new VMLs which fit to the mental representations of their users? Are they easy to use by VML designers? How they correspond to other methods of evaluating VMLs?

This paper describes an application of cognitive dimensions to evaluation of a visual modeling language during the construction of the language. This work is a part of MAYDAY EURO 2012 project which deals with providing technology for efficient multimedia stream processing in a supercomputer cluster environment. There was a need for a new language to facilitate the work of application developers by elevating the level of abstraction in composition of complex service structures from simple services and to enable automatic generation of executable service structures for KASKADA platform [5] which is a middleware solution supporting multimedia stream analysis. The specifics of processing multimedia streams by services which are parallel, continuous and pipeline-style and which are executed in the high-performance supercomputer environment means that existing modeling languages [1, 3, 7, 8] are inadequate for this application. Therefore we have decided to develop a kind of domain-specific language, which has been called multimedia stream processing modeling language (MSP-ML). In fact, there was a need for a modeling solution which included the language and a tool for performing round-trip transformations to executable service structures integrated with the KASKADA platform.

Evaluation with cognitive dimensions was a part of a systematic method of engineering visual modeling languages. The method included the analysis of requirements and expectations of MSP-ML and models created with this language in their context of use. The evaluation was preceded with 'VML design with quality in mind' which meant that VML designers have considered the quality criteria of conceptual expressiveness, visual expressiveness and consistency with the tradition of modeling when making the decisions about elements of the language and their notation. MSP-ML was described using the notation and rules of OMG Meta-Object Facility standard [6]. Apart from applying cognitive dimensions for evaluation of the usability of the language, several tests with automatic transformations were applied to verify conceptual expressiveness for transformations. Furthermore, empirical studies with potential users of the MSP-ML were conducted in order to check intuitiveness of the notation as well as the level of its acceptance by its users.

The paper is structured as follows. Section 2 contains the introduction to the language and describes the method of evaluation with CDs. Section 3 presents the results of evaluation. Section 4 describes the comparison of cognitive dimensions with methods using other criteria for VML evaluation. Section 5 draws conclusions.

2. MSP-ML and CD-VML-MSP

MPS-ML allows for modeling complex service structures from simple services which represent simple algorithms. An example of the diagram made with MSP-ML, which models car plate detection service, is presented in Figure 1. Empty circles represent inputs, filled circles - outputs. Rectangles with two sharp and two rounded corners represent simple services with their input and output pins. All elements with dashed lines are related to messages while others are related to multimedia streams. The arrows represent channels: a thick arrow is a multimedia stream channel and a dashed arrow - message stream channel. The constraints in the middle part of a simple service represent service parameters, such as execution or quality parameters.

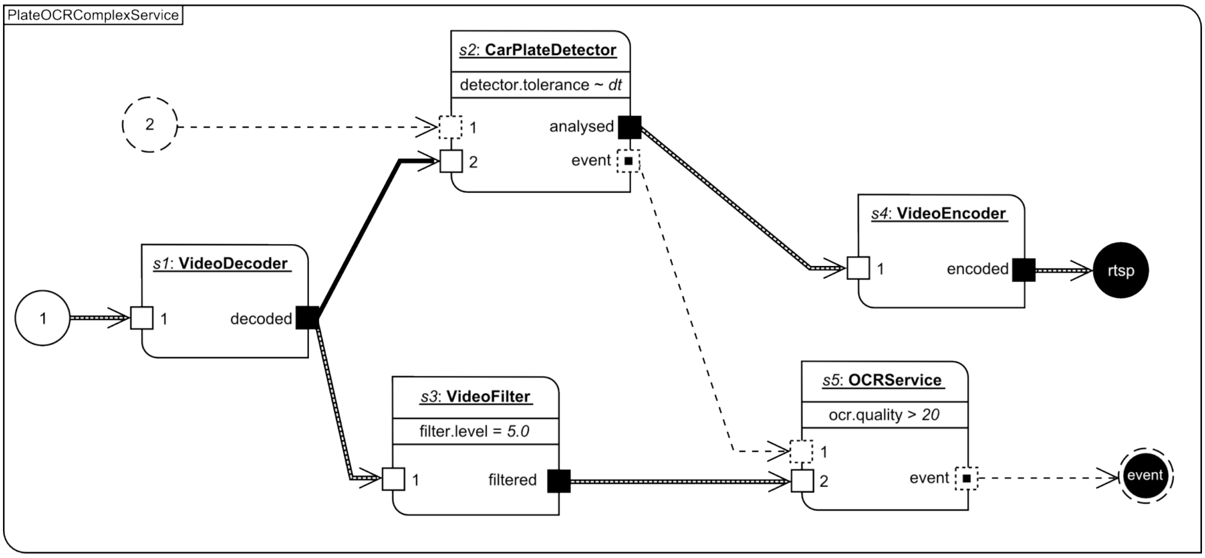


Figure 1. An example of a diagram made with MSP-ML for car plate detection.

CD-VML-MSP method is a customization of the original cognitive dimensions for visual modeling languages [2] in a version for multimedia stream processing language in the context of use for modeling and automatic round-trip transformations between diagrams and executable service structures. A template of the CD-VML-*XX* method (where *XX* should be replaced by the name of the VML after customization of the template) contains several questions which capture advances in visual modeling language engineering and relates them to the elements of information representation and processing by humans. They allow examination of the details of VMLs against cognitive aspects defined by the cognitive dimensions. For example, the definition of closeness of mapping is closeness of language mechanisms to the natural mental representations of domain. The motivation is to identify problems with closeness of mapping. System models should be as close as possible to the system perception, and modeling mechanisms should be as close as possible to mental representations. The following questions are related to this dimension:

3.1 How closely related are models to the real system perception?

3.2 Is there a clear correspondence between model elements instances and parts of the system?

3.3 How closely related are combinations of the model elements and modeling mechanisms to the concepts and basic activities you think with?

3.4 Which notation elements do not fit to the concepts you think with or model elements they represent?

3.5 Can you think about any constructions that seem unnatural or artificial and hard to understand?

3.6 Which model elements or notation elements are they related to?

3.7 Which modeling mechanisms or visual mechanisms seem to introduce additional difficulties or cause that something is 'strange way of doing things'?

3.8 If there are different stakeholders using the model, is it easy and natural for all of them?

The CD-VML-MSP method can be used for evaluation by several users for focusing their attention. In order to systematically conduct the evaluation, tables containing all model elements should be prepared before evaluation. An example of a fragment of such a table is presented in Table 1. As the question asks about model elements, notation elements and concepts, the table contains both textual and visual representations with the space for the description of problems related to them in context of the question. The example contains only pins and channels, the full table contains all model elements and their notation elements.

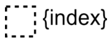
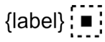
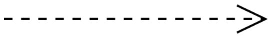
Model element	Notation element	Problem description
Multimedia Input Pin		
Multimedia Output Pin		
Message Input Pin		
Message Output Pin		
Multimedia Channel		
Message Channel		

Table 1. A fragment of CD-VML-MSP questionnaire which facilitates answering a question related to closeness of mapping asking 'Which notation elements do not fit to the concepts you think with or model elements they represent?'

The team which performed MSP-ML design including evaluation with CD-VML-MSP has consisted of three persons. One of them has VML design expertise and a good knowledge of cognitive dimensions. The second is a specialist in domain including application developers' expectations, KASKADA platform and its supercomputer environment. The third was the main developer of the tool for round-trip transformations integrated with the KASKADA platform. The resulting language was created in close cooperation. Practically, the team member with the expertise in VML design and cognitive dimensions has leaded several meetings by means of selecting topics, explaining details of a given method, moderating discussions about the use of this method in this case, formulating remarks and gaining acceptance of other team members. This way all elements of MSP-ML were evaluated against all questions related to cognitive dimensions. We have observed that team members were not interested in the details of VML theory unless they influenced the language design or were a reason for explaining any decision. This was in case of evaluation MSP-ML with cognitive dimensions as well as application of other elements of the theory of VML design. The team members tended to move forward instead of discussing more and more details. Thus, the role of VML design expert was to facilitate design with several methods from a VML design toolbox including CD-VML-MSP and sustaining discussion until reliable information was discovered.

3. Results of Evaluation

Precision allows to check the precision of description and consistency in different parts of distributed description. In case of MSP-ML, the precision was evaluated as high. The focus on conceptual expressiveness resulted in very detailed distinctions between the model elements. Each of the model elements has a precise meaning and a clear mapping to a fragment of executable service structure file.

Role expressiveness allows to discover the level of understanding for purpose, scope and importance of the existing mechanisms with the motivation to check out whether modeling mechanisms or model elements are what users need - whether they have a clear purpose, scope and meaning and there are no redundant or missing mechanisms. When examining MSP-ML according to this dimension, no redundant or missing mechanisms has been discovered. The language seems to provide just necessary mechanisms for modeling.

Closeness of mapping allows to focus on closeness of language mechanisms to the mental representations of domain. In case of MSP-ML, the closeness of mapping of channels and pins was estimated as high. The multimedia streams have a thicker icon comparing to message streams and this is intuitive because multimedia streams contain more data. Input pins and streams are modeled as empty elements of notation (before processing), while output pins and streams are modeled as filled elements (after processing). The most controversial element is the representation of services which have characteristics of both structure elements and dynamism of continuous stream processing.

Consistency allows to check whether similar concepts are expressed by similar model elements and similar model elements have similar notational elements in a given context of use. In case of MSP-ML, the consistency criterion is satisfied. All pins are represented as small rectangles. All external elements such as inputs and outputs modeled as circles and channels represented as arrows. The second sub-dimension of consistency is concerned with the distinction between a multimedia stream and a message stream. It is also satisfied, as all message-related constructs are represented with dashed lines (in the final version of the language). Finally, the abovementioned distinctions between inputs as empty elements of notation and outputs as filled elements are set up consistently as well.

Diffuseness allows to search for mechanisms that cause verbosity of language usage with the motivation to identify descriptions that take too much space and their reasons. Descriptions that are too long are not efficient and they may cause mistakes. No problems with this dimension were identified.

Visibility allows to focus on mechanisms that language delivers for information integration, mechanisms that allow to see views looking at notation elements, and model – looking at diagrams. The motivation is to identify views that should be easy to find looking at diagrams. There should be good visibility of views important while modeling and using models. In case of MSP-ML only one view from user's perspective is needed and it is supported.

Abstraction allows to identify available and useful abstraction mechanisms. Thinking at higher level of abstraction may be more difficult and cause imprecision or ambiguity, but on the other hand, it allows to deal with larger problems. Abstraction mechanisms should allow for describing system at different levels of abstraction. In case of MSP-ML, the goal was to elevate the level of abstraction. Instead of long XML files which contain many details and do not show relationships between the simple services, MSM-ML allows to eliminate the number of redundant information and explicitly present relationships between the services. Thus, this goal is satisfied.

Hidden dependencies dimension aims at identifying mechanisms which include hidden dependencies, i.e. elements which are not visible although they should be. Hidden dependencies may cause problems while using and changing models. No hidden dependencies in MSP-ML language were identified.

Provisionality allows to focus on mechanisms that allow for expression of concepts that are not complete and they can be useful to sketch solutions. The motivation is that model development is an iterative activity and, usually, first sketches of models are created and then, the details are added. Provisionality can be useful for modeling simple services without parameters. The simple services and their relationships seem to be more fundamental part of models than details of parameters, thus a simple service without parameters should be a syntactically correct construction. The realization of this idea should be supported by the tool.

Secondary notation deals with mechanisms that allow to extend the language. The motivation is to identify the need and availability of mechanisms that allow to specify things that are not possible to express explicitly using language elements. The language is quite simple, completely defined for a particular domain and thus, no need for such mechanisms was identified.

Viscosity focuses the attention on mechanisms that can cause difficulty in introducing a change with motivation to view the language from the perspective of changing models. As MSP-ML is based on basic model elements which correspond to entities in the KASKADA platform, e.g. simple services with their parameters and pins, inputs/outputs related to complex services; and relationships are built upon the basic elements, no potential problems with viscosity were identified.

Hard mental operations should allow to indicate mechanisms that require high demand on cognitive resources in order to identify things that can be difficult to do because they can cause errors and decrease in efficiency. No potential problems with hard mental operations related to the language were identified. The most difficult part of modeling is supposed to be the task of matching services, which is not the responsibility of the language, but decisions made during modeling. The tool probably could help with presenting details of services.

Error-proneness aims at identifying mechanisms that invite mistakes with motivation to find typical errors and their reasons. No error-proneness mechanisms has been identified. However, they may appear when modeling is done by application developers. The tool for modeling can prevent mistakes by providing contextual help or blocking syntactically wrong relationships on diagram.

Premature commitment is defined as dependencies between mechanisms that force order of activities. The motivation is to identify activities with unnatural order of doing things because it can cause several mistakes or difficulties, e.g. those that force you to think ahead or make certain decisions before you can model what you want. No problems with this dimension were identified.

The above analysis has confirmed that the decisions made when designing MSP-ML were right in most of the cases and no serious problems with the language were identified. Additionally, this evaluation allowed for formulation of a few new suggestions for the tool. The impression of the involved VML designers was that some of the cognitive dimensions appeared as very important, e.g. closeness of mapping, while others - less useful, e.g. viscosity, premature commitment. However, in case of evaluation of other VMLs, this situation could be different.

4. Comparison of the Results with other VML Quality Criteria and Methods

As our method of VML engineering included also 'design with quality in mind', tests with round-trip transformations between diagrams and executable service structures and empirical studies with potential users, it is possible to compare the results of application of cognitive dimensions with the results of application of those methods.

The design with quality in mind was about considering the following VML quality criteria when making decisions about VML design:

- conceptual expressiveness - the VML should allow for modeling all required concepts and constructions from both user's and automatic transformation's points of view;
- visual expressiveness - the notational elements should intuitively indicate the meaning they convey;

- consistency with the tradition of modeling - the elements of models and notation should be consistent with popular modeling languages, e.g. the icons used in UML.

The conceptual expressiveness is related to cognitive dimensions such as precision, role expressiveness, conceptual aspects of closeness of mapping and conceptual aspects of consistency. The visual expressiveness can be related to visual aspects of closeness of mapping and visual aspects of consistency. The consistency with the tradition of modeling may be one of the reasons why closeness of mapping or consistency is satisfied.

During the VML design, several discussions have led to extending provisional representations and increase the conceptual expressiveness by adding new language elements. Later, the tests have confirmed, that from the automatic transformations' point of view, this quality criteria is satisfied. The evaluation with cognitive dimensions have not indicated any serious problems with the related dimensions.

Empirical studies with MSP-ML users were conducted in order to verify intuitiveness of the MSP-ML and the level of acceptance of the notation by application developers. The following description is intended to give the reader an idea of the empirical studies to allow understanding comparison to results of evaluation of cognitive dimensions. In the studies participated sixteen potential users of the language, who have known the technology and had a need for such a language but they have not seen MSP-ML before. The first part of the study aimed at checking intuitiveness of the notation. Participants received a diagram in MSP-ML and they were supposed to recognize model elements. As 95% questions were answered correctly and twelve participants (75%) have answered correctly to all questions, the intuitiveness of the language was confirmed. The second part of the study aimed at exploring fitness of the notation to preferences of application developers. It contained several suggestions of elements of notation for each model element. Furthermore, the application developers could propose an alternative symbol if none was adequate in their opinion. They were encouraged to express their opinions independently of the first part of the studies. Our knowledge of participants personality profiles and the large number of several suggestions they expressed allow to reason that the results of the second part of the studies were not significantly biased by the first part of the studies. In cases, when the decisions about notation elements were quite obvious for VML designers, the large number of participants were choosing the notation elements as the most suitable. In cases, when VML designers had to make a decision among propositions having contradictory arguments, only some of the participants were selecting that notation elements while others preferred other options. For example, in case of message stream channel, the majority of participants have selected dashed arrow and VML designers decided to make a change in the language. When designing the language, internal consistency as a part of visual expressiveness (all message related elements modeled with dashed style) was in contrast with consistency with tradition of modeling (modeling message as a simple arrow). The participants of the studies have indicated internal consistency as more important.

5. Conclusions

Usability and cognitive fit are perspectives that are difficult to capture. Regarding the questions whether CDs are useful and easy to use by VML designers for constructing new VMLs, we have observed the following. Having the CD-VML-MSP method which asks several questions related to this area was helpful. It allowed to focus attention on several aspects related to mental representations and working with the language in a step-by-step manner. It helped to drive discussions and structure results description. The application of the method gave a sense of having more 'professional' approach comparing to just considering several VML quality criteria. MSP-ML is quite simple and the application of CD-VML-MSP method was considered as complicated and time-consuming. As we have mentioned before, the team members tended to finalize discussions with decisions regarding the language rather than discussing the details of impact of cognitive dimensions on the language elements. In this case, the evaluation has mainly confirmed that the decisions about VML were right. It provided several views and arguments about the language. However, we observed that any method cannot replace creativity and decision making. In the context of constructing the language, it is not enough to have several, sometimes contradicting, arguments. The focus is on making decisions what the language elements should be. Cognitive dimensions was useful as a method of evaluation, but other methods for driving the design were also needed.

Regarding the question about the correspondence of the results of evaluation with CDs to other methods of evaluating VMLs, in our opinion, several VML quality criteria or methods of evaluation deliver complementary results. The design with quality in mind surely led to construction of a better language than it would coincidental design. The evaluation with CD-VML-MSP has mostly confirmed that the decisions were right which probably would not be the case with coincidental language design. As we have discussed in the previous section, quite often the terms indicating for quality criteria in different conceptual spaces have overlapping meaning and it is difficult to say which set of terms is most suitable in general. Furthermore, in our opinion some relationships exist between concepts from different conceptual spaces, for example, a relationship between language intuitiveness and tradition of modeling and knowledge of a VML by an individual; or a relationship between visual expressiveness and satisfaction of developers. The exploration of these relationships in general requires more studies.

Finally, one could ask a question whether application of cognitive dimensions (as a kind of analytical method) could replace the empirical studies with participation of potential users. As our experience has shown, the involvement of users is very valuable and these methods deliver complementary results. Empirical studies allow to capture several subjective and psychological aspects which have impact on language acceptance and satisfaction from modeling. Moreover, potential users deliver a lot of interesting ideas which can be used in order to improve the language. They can also indicate for user preferences when facing contradicting arguments. Thus, when users are available it is worth to make them involved in a short empirical study anyway.

6. Acknowledgements

The work was partially supported by MAYDAY EURO 2012 project, Operational Program Innovative Economy 2007-2013, Priority 2 “Infrastructure area R&D”.

7. References

[1] Bell M. (2008) Service-Oriented Modeling: Service Analysis, Design, and Architecture. John Wiley and Sons.

[2] Bobkowska A.(2005), A Methodology of Visual Modeling Language Evaluation, Proceedings of SOFSEM 2005, LNCS 3381.

[3] Object Management Group (2011), Business Process Model and Notation (BPMN), www.omg.org

[4] Green, T. R. G. (1989), Cognitive dimensions of notations. People and Computers. Cambridge University Press. Cambridge 443-460.

[5] Krawczyk H., Proficz J., KASKADA – multimedia processing platform architecture (2010), In: SIGMAP 2010, Proceedings of the International Conference on Signal Processing and Multimedia Applications.

[6] Object Management Group (2006), OMG Meta-Object Facility v.2.0, www.omg.org

[7] Object Management Group (2010) OMG Unified Modeling Language v.2.3, www.omg.org

[8] Object Management Group (2009), Service oriented architecture Modeling Language (SoaML) - Specification for the UML Profile and Metamodel for Services (UPMS), v.1 - Beta 2, www.omg.org