Suitability of Visual Modelling Languages for Modelling Tangible User Interface Applications

Eric Tobias, Eric Ras
Service Science & Innovation Department
Public Research Centre Henri Tudor
J.F. Kennedy av. 29, L-1855 Luxembourg, Luxembourg
Email: {eric.tobias, eric.ras}@tudor.lu

Nuno Amálio
Faculty of Science Technology and Communication
University of Luxembourg, Luxembourg
rue Richard Coudenhove-Kalergi 6, L-1359 Luxembourg
Email: nuno.amalio@uni.lu

***Abstract*—This paper compares the suitability of visual modelling languages for describing tangible user interface (TUI) applications. After gathering different approaches, we have selected three languages for our comparative study: the visual object constraint language (VOCL), augmented constraint diagrams (ACD), and the visual contract language (VCL). A weighted evaluation based on multiple quality criteria led to the conclusion that VCL is best suited to model TUI applications.**

***Index Terms*—Visual Languages; Model Driven Development; Comparative Study; VOCL; Constraint Diagrams; VCL**

I. Introduction

There is considerable hype surrounding domain-specific languages (DSLs), which is an attractive idea that has proved useful for certain problems. However, this hype can often be misleading: one may be tempted to invent a new DSL rather than trying to find a suitable general-purpose language.

This paper evaluates the suitability of general-purpose visual modelling languages (VMLs) to the design of tangible user interface (TUI) [1] applications. This is part of our research work on modelling a TUI application. TUIs manipulate a digital world using physical devices and projections to take advantage of known metaphors of the physical world. Our TUI application consists of an environment for building Business Process Modeling Notation 2.0 (BPMN2) diagrams.

II. Inclusion and selection criteria

We must determine *what* to compare and *how* to compare. The how is addressed in section III. The *what* or the suitability was determined by the following criteria:

- The languages need to be visual. This is an appropriate medium for the scope of TUI development. The language must be able do describe visually all properties of interest. Another strong criteria is the need for a modelling rather than a programming language.
- The study only considers general-purpose languages. A feature of TUIs is their strong interaction with the real world that often needs to be modelled. We therefore need languages that are capable of dealing with this diversity and that are capable of describing modelling requirements across many fields, yet specialised languages might only be suited for a specific task. Furthermore, a more general and abstract nature enables the language to use metaphors and abstractions to draw upon existing language schemata and improve language learning [2]. As noted by Schema Theory [3], the ability to understand a new domain can be improved by reusing familiar concepts in already existing schemata.

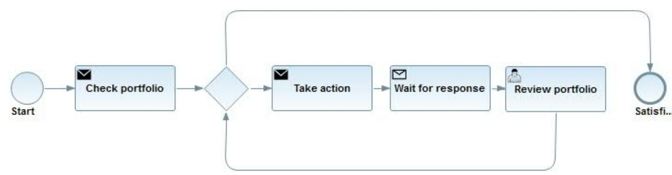

Fig. 1. BPMN2 scenario.

III. Study

From the thirty one gathered resources, only roughly half remained after pre-selection. Those were then further grouped and distilled to identify three major contestants:

- UML2 with the visual object constraint language (VOCL) [4].
- Augmented constraint diagrams (ACD) [5].
- The visual contract language (VCL) [6].

To perform the comparison, a simple BPMN2 scenario was devised (Fig. 1) and the relevant requirements extracted. These requirements were then modelled with each VML to gather data on their performance. The scenario and each modelling endeavour were undertaken by one of the authors to counteract the effect of familiarity, skill and understanding of the task on the result. It is to note that the author doing the modelling had some experience in all chosen VMLs apart from ACD.

The measurements categories collected after each modelling step are summarised in Table I. Each measure was attributed points according to Table II. Further details on measurements categories and scoring scheme can be found in the accompanying technical report [7].

IV. Results

The study's scores were marked as by the scoring scheme and subsequently weighted. The weighting, being the result

[1]Where X is either structural requirements, behavioural requirements or constraints.

Tool support	Availability Maintained Latest version Branch
Semantics & Transformation	Formally defined Transformability
Expressivity	# [X][1] # [X] satisfied # requirements partially satisfied # unsatisfied requirements Ratio
Usability	Naming conventions Naming fit Documentation Tutorial Hands-on-tutorial Primitive mutability Live suggestions
Error Checking	Time Syntax highlighting Degree Error correction suggestion Debugging possible
Validation & Verification	Modularity Verification scheme

TABLE I
MEASUREMENT CATEGORIES

10	Desirable
5	Manageable
2	Acceptable
-5	Undesirable
-∞	Disqualification criterion

TABLE II
SCORING SCHEME

Version	3.2506.1044	UML2 + VOCL	Augmented Constraint Diagrams	VCL
Tool support				
	Availability	10	10	10
	Maintained	-5	-5	5
	Latest version	10	10	10
	Branch	5	5	5
Expressivity				
	# Structural requirements	27	27	27
	# Behavioural requirements	11	11	11
	# Constraints	4	4	4
	# Structural requirements satisfied	10	10	10
	# Behavioural requirements satisfied	10	10	10
	# Constraints satisfied	10	10	10
	# Requirements partially satisfied	0	0	0
	# Unsatisfied requirements	0	0	0
	Ratio	10	10	10
Semantics & Transformation				
	Formally defined	2	10	10
	Transformability	2	2	10
Usability				
	Naming conventions	10	10	10
	Naming fit	10	5	5
	Documentation	5	10	5
	Tutorial	2	2	2
	Hands-on tutorial	2	2	2
	Primitive mutabiltiy	5	5	5
	Live suggestions	2	2	10
Error Checking				
	Time	-5	-5	10
	Syntax highlighting	5	-5	10
	Error checking	2	-5	2
	Error correction suggestion	2	-5	2
	Debugging possible	-5	-5	-5
Validation & Verification				
	Modularity	10	5	10
	Verification Scheme	2	2	2
Weighted total		18,99212598	15,77165354	26,92913386

TABLE III
WEIGHTED RESULTS

of a questionnaire sent out to a test sample of practitioners proved to not affect the outcome significantly. See [7] for further details. The evaluation scores for this small study can be seen in Table III. The results show that VCL surpasses ACD due to the latter's non-existent tool support, which makes error checking tedious. The combination of UML2 & VOCL looses ground on tool support, when compared to VCL, and formalism, due to lack of formality for some parts of UML2.

V. CONCLUSIONS

The study presented here is a preliminary evaluation done with the purpose of choosing a VML for modelling a TUI application. To the authors knowledge, this is the first study comparing this particular set of VMLs. Although the evaluation has been done using a small case study, the results are interesting giving a good overall picture on the state of the art in visual modelling approaches. The fact that three general-purpose modelling languages are, at first sight, suited to model TUIs is encouraging. For this small case study, all evaluated VMLs were capable of describing the requirements. Some VMLs have seen little development since their inception, after providing a proof of concept. It was VCL, the most recently developed VML and hence still actively being support and featuring modern model-driven facilities, that showed to provide an advantage over the other contestants. This advantage may, however, vane with the conclusion of the project. A future goal of being able to visually model applications directly using VCL in a collaborative TUI environment is not unreachable if VCL continues to be supported.

REFERENCES

[1] O. Shaer and E. Hornecker, "Tangible user interfaces: Past, present, and future directions," *Foundations and trends in Human Computer Interaction*, vol. 3, no. 1–2, pp. 1–137, 2009.

[2] F. Boers, "Metaphor awareness and vocabulary retention," *Applied linguistics*, vol. 21, no. 4, pp. 553–571, Dec. 2000.

[3] S. A. Widmayer, "Schema theory: An introduction," *Retrieved April*, 2005.

[4] K. Ehrig and J. Winkelmann, "Model transformation from visual OCL to OCL using graph transformation," *ENTCS*, vol. 152, pp. 23–37, 2006.

[5] J. Howse, S. Schuman, G. Stapleton, and I. Oliver, "Diagrammatic Formal Specification of a Configuration Control Platform," *Electronic Notes in Theoretical Computer Science*, vol. 259, pp. 87–104, Dec. 2009.

[6] N. Amálio and P. Kelsen, "Modular design by contract visually and formally using VCL," in *VL/HCC 2010*, 2010.

[7] E. Tobias, E. Ras, and N. Amálio, "VML Usability for Modelling TUI Scenarios - A Comparative Study," University of Luxembourg, LASSY, Tech. Rep. TR-LASSY-12-06, 2012, available at http://vcl.gforge.uni.lu/doc/VMLCaseStudy.pdf.