

Change Report

In the following, I have collected the review comments, numbered them, and show what I have done to address them. I omit in this enumeration the various typos and style-improvements as remarked by the reviewers; of course, these have been addressed, too.

ID	Reviewer	Remark	Action taken
1	1	- The relation between VMQL and MOCQL. At some point you basically say that MOCQL is a textual syntax for VMQL. This needs to be clarified. Is MOCQL really a new language or just an alternative concrete syntax for an existing one?	MOCQL is a new language using concepts first studied in VMQL. Think of C++ vs. Smalltalk. I have clarified this in various places, I think I have found all formulations that were in danger of being misinterpreted.
2	1	- I don't really understand why the strong emphasis on VMQL in the text (specially if the answer to the previous question is that MOCQL is an independent language). Clearly, practitioners don't care about VMQL (I don't pretend to say this in a pejorative way but I guess you agree with me that it's not exactly a popular language) so the comparison should be only with OCL.	My goal is not to make MOCQL (or VMQL) popular: they are research vehicles that shall demonstrate that we can do better than OCL. This is a thread of research that tries to produce the best possible solution, and I am getting closer with each step. I mention VMQL not to promote it but to document the evolutionary process.
3	1	- There are a lot of contradictory messages. You say you proved that VMQL was a great language but now you say that MOCQL is much better? Strange. Also, in your experiments, OCL seems to do quite well in comparison with VMQL (even better sometimes) which is not what you were saying at the beginning.	It turns out there is a typo in the labels of the table: OCL and VMQL should be the other way round. The graph was correct, though.
4	1	- Apart from measuring usability you should compare the languages at other levels. Expressiveness obviously, but also, what about writing MOCQL expressions? I'd say that something that it is easier to read does not necessarily imply that it's also easier to write. Maybe OCL has a more steep learning curve but then people are more productive with OCL than with MOCQL.	You're right in demanding a more comprehensive usability evaluation that includes writing – this is ongoing work that I cannot yet publish, since our study does not yet allow meaningful interpretation (too small n on this task).
5	1	- Does MOCQL work with non-UML models?	Conceptually: pretty sure – but I haven't implemented/validated this yet.
6	1	To sum up, I liked the fact the author tries to create a better query language but I don't think he does a good job in neither convincing us about the qualities of the language nor precisely defining the new language status and its relationship with the othhres.	In a conference paper, there is not enough space to discuss all qualities, and this is not the focus of this paper anyway.
7a	2	I suppose MOCQL is mainly aimed to query large models. In fact, model querying is necessary (and crucial) for managing large models, but this issue is not commented in the paper.	Correct – on both counts. I added this in the introduction.
7b	2	The author should include any comment about model	Thanks for this remark – that one is really helpful!

		repositories (e.g. CDO and Morsa) in the introduction of the work. He should to indicate how the approach proposed can be used from these repositories	I included references to model repositories when I explain the overall approach.
8	2	- The author only considers one of the possible dimensions to be considered in comparing GPLs and DSLs, but the successful of MOSQL not only depends of usability, although this factor is essential. Some classic metrics could have been measured to give some empirical evidences on the facility to learn the language or the verbosity of the grammar (Crepinsek, M., et al. (2010). On Automata and Language Based Grammar Metrics. Journal on Computer Science and Information Systems, 7(2), 310-329.).	I can't address this in the current format, that would be a paper in its own right.
9	2	- The first proposal of the author on the translation of models to Prolog was presented in 2007. I do not understand the lack of a performance evaluation, especially when the author states in section 6 as future work that: "we will have to demonstrate that the performance of MOCQL can be at least as good as that of existing OCL tools". Scalability is an essential issue for the adoption of a model query language.	see above and answer to comment 15c
10a	2	- I had problems to understand that MOCQL is a language applicable to any metamodel, not only to the UML metamodel. I think my confusion was justified and caused by the examples and the lack of a clear statement on this issue in the paper. In the Abstract is said that "...MOCQL is a declarative textual language to express queries for UML models"; the sample models are an UML class diagram and an UML use case diagram; queries such as "find all classes ..." or "find all abstract classes ..." suggest to the reader that MOCQL is only applicable to UML.	I clarified this in the introduction and added a section 4.6
10b	2	In addition, the model-to-Prolog translation is not enough explained.	I am not sure exactly what kind of explanation you are missing.
11	2	- Therefore, I think that Section 2 should be rewritten to change the example by a non-UML model, for instance a simple state machine or Petri net, or a simple process model. In addition, the author should make clear to readers if the query is applied on the metamodel (M2 level) or the model (M1 level). The metamodel of the model example should be shown in a figure. It should be indicated and commented the more significant types of query (find, count, save, etc), and how they are built, for	I think that using UML as an example is preferable to using BPMN or EPCs since UML contains 14 notations, one of which covers the same domain as BPMN and EPCs, and with a much higher complexity, too. So, if MOCQL can handle, UML, it can trivially handle BPMN and EPCs. See also comment 10a

		instance the reference to elements of the metamodel (“classes”, “actors”, “activities”,...); how navigation can be expressed, etc.	
12	2	In short, with regard to the previous work of the author, the main contribution of this article is the experiment for evaluating the usability of the languages, since that the MOCQL language is superficially presented. However, I think a detailed description of the capabilities of MOCQL is needed to assess the contribution of the language.	The main contribution is a new model querying language. I claim higher usability than OCL, and I provide evidence for that claim. Any other concerns are not the subject of this paper.
12b	2	A comparison of MOCQL with OCL can not only consider the usability, but it is necessary compare other aspects such as the expressiveness, completeness and safety.	This is beyond the point: the goal of this paper is not to compare MOCQL and OCL, but to promote usability of model querying languages.
12c	2	In addition, the proposal lack of a performance evaluation.	see above
13	2	Abstract: Remove the restriction to UML models	done
14a	2	- “VMQL is more easy to use than ...” should be changed to “VMQL is easier to use than ..” - “(VMQL) more or less shares the conceptual ...”, you should be more precise.	done
14b		- “.. was even more popular ...” more popular? How have you measured this?	Yes, one of the questions on the questionnaire asked for user preference – but I say this in the very same sentence, so I don’t quite understand the remark. I have reworded the phrase, though, hoping to better avoid misunderstandings.
15a	2	- “The way associations are represented in UML and XMI would make ...” I am not able to understand how the expressiveness of OCL is affected by XMI.	I removed the misleading reference to XMI.
15b	2	- The grammar is not commented, you only includes examples for “find”. In such case, this figure covering 1 page is unnecessary.	I don’t understand what you say. Why would I comment the grammar.
15c	2	- You should include some study on performance of using Prolog to query models (Querying UML Models using OCL and Prolog: A Performance Study, Joanna Chimiak_Opoka et. al) and the usage of logical formalisms (e.g. Maude) to represent and query models.	Updated Section 5 (“Related Work”) accordingly. The reviewer might be interested to read the “Acknowledgements” section of that paper.
16	3	- The language seems to be rather incomplete: the author do not seem to realise the full spectrum of the expressiveness of his own language, which I agree to be quite reasonable at this stage of the language design.	no action taken
17a	3	- Miss a discussion about the problems identified with the	I’d rather cut out this aspect to improve the focus of the paper. If you’re

		comparison with the visual language VMQL, and perhaps other visual languages.	alluding to visual OCL: a meaningful comparison from a usability perspective will require another set of experiments and is beyond this paper.
17b		The results were not so convincing as happened with OCL.	see comment 3.
17c	3	Why not to choose a visual concrete syntax for the language?	You are referring to Visual OCL, I think, and that it is just as bad as OCL. The point is that there are (at least) two different factors at play: the concrete and the abstract syntax, and the abstract syntax is more important, it seems. The third factor is the way the questions are asked, but that's a different paper...
18	3	- No translation to OCL was provided. Since the comparison was against OCL, the author should provide a complete translation to OCL (instead of to a rather incomplete translation to prolog) in order to better assert the MOCQL's expressiveness.	I think this is both unhelpful and way beyond the scope of this kind of paper.
19	3	- Not enough references that indicate where did the author decided to perform such kind of usability evaluation (perhaps in user interfaces' usability evaluation?). These references can be very valuable for other language designers to guide them while performing such kind of evaluations.	Not sure what you mean by "where did the author decide" – do you mean "why"? Then the answer is: personal experience showed me how difficult to use OCL is – I have added that in the abstract now.
20	3	Figure 6 is totally redundant with table 2	True, there is no new data in Fig. 6 that isn't already in Table 2. However, I believe that visualizing these data is important for the readability of the paper to provide this graph, justifying this redundancy, in particular in the presence of remark 3. I amended the caption of Fig. 6 to avoid confusion. I have also added the std. deviations to Table 2.
21	4	The reviews do point out that the clarity of the text is still problematic at some points. In particular, the three independent reviewers expect a more formal treatment of the relation between MOCQL and OCL.	There is no relationship between MOCQL and OCL other than that they can be used for the same purpose.
22a	4	The first reviewer for example is not yet fully convinced about the paper as he expected mainly a comparison of MOCQL with OCL w.r.t. to all aspects like also expressiveness. Arguably, this expectation is too high.	I hope I now clarified this misconception in the introduction: comparing these two languages is not the objective of the paper.
22b	4	Nonetheless, the following key points should be resolved for the final version, otherwise the paper will not be accepted: Does MOCQL work with non-UML models?	We haven't tried yet. We expect a few minor conceptual adaptations to apply MOCQL to BPMN (ongoing work), and certainly we will need to change parts of the implementation, but apart from that, MOCQL should apply to any modeling language whatsoever, including DSLs.
23a	4	Please also address the R2 comments (cf., "I had problems to understand that MOCQL is a language applicable to any meta-model...") in this context.	see above

23b	4	R3 suggests to provide a small non-UML example.	That would be purely hypothetical, and unless I have evidence that this works (i.e., an implementation), I won't raise any such claim.
24	4	Also the second reviewer expects a comparison of MOCQL with OCL w.r.t. to several other aspects. The author gives some hints about differences. Arguably, these points go beyond the research objective for this paper but the author should take into account that the independent reviewers did get these expectations after carefully reviewing the submission. Therefore, we stress that the final version should present the scope of this work more explicitly and argue that this "small" problem is still interesting.	In order to better explain the scope, I have rephrased parts of the abstract, introduction, and Sections 6.2 and 6.3 ("Contributions and Limitations" and "Future Work") to address this, and added a new subsection 3.6. I have highlighted the ideas leading to and pursued in this research in order to make it clearer what the contribution of this paper is.
25	4	The third reviewer even asks for a complete translation of MOCQL into OCL. Although arguably it was not his intention to illustrate the expressiveness compared to OCL it is clear that the three reviewers do expect some more elaboration along these lines. The author should ensure that readers will not have a similar "reaction" when reading the final version of the paper.	I concur: it is not possible in this paper to present a complete translation, see also remark 2. I have updated the Introduction in an attempt to prevent any misunderstandings along these lines.
26	4	the paper clarifies better how MOCQL can be applied to non-UML models. It should also be clear whether this support is theoretical, or already supported by a generic tool.	done, see Abstract, "Introduction", and "Discussion" which have been restructured and partially rewritten.
27	4	the introduction should be more precise and *convincing* about the scope of this work. In particular, the introduction of the final version should convince the reader that the incompleteness w.r.t. research on the expressiveness etc. does not compromise the results of the evaluation.	By better focusing the paper and making the objective clearer, I hope that this remark is also being addresses. You'll be the judge on whether I'm more convincing this time! ☺

In summary, I have rewritten the Introduction and the Discussion, changed many pieces along the way, and I have added a section 4 on Implementation – this is necessary to show the overlap/difference to VMQL and, what it would take to extend MOCQL to languages other than UML. In this context I would like to refer the reviewers to a paper we have just now submitted to BM-FA, a workshop co-located to ECMFA. Here, we show how VMQL – in many ways the predecessor of MOCQL – can be extended to BPMN.