

# The Future of Model Transformation Languages: An Open Community Discussion

Loli Burgueño<sup>ab</sup>    Jordi Cabot<sup>ca</sup>    Sébastien Gérard<sup>b</sup>

a. IN3, Open University of Catalonia, Spain

b. Institut LIST, CEA, Université Paris-Saclay, France

c. ICREA, Spain

**Abstract** Model transformations are the key element that brings life to model-driven engineering. Animation, simulations, VV, code-generation, etc. all depend on some kind of model transformation to work.

Model transformations are typically defined via specialized model transformation languages but this is now in question due to the lack of convincing evidence that specialised languages are substantially better than general-purpose languages for model transformation specification, and the rise of artificial intelligence. We report on the results of an open discussion with the model transformation community on the future of these languages, including whether such a future exists at all.

**Keywords** modeling, model transformation, domain-specific language, survey.

## 1 Introduction

Model-Driven Engineering (MDE) is a software development methodology in which models are first-class citizens driving the life of the project. These models need to be manipulated—for instance, to generate code, perform analyses or for data integration—and model transformations are the key element to achieve this.

The traditional way to tackle model transformation problems is to write a transformation program using a dedicated model transformation language (MTL) such as ATL [JABK08], QVT [OMG05] or ETL [KPP08]. Although many different and mature model transformation languages exist [SC12], they have never been massively adopted yet. Indeed, model transformation languages are popular in academia but, in practice, many companies still prefer to write their transformations directly in general-purpose languages like Java—probably, because there is no convincing evidence that dedicated MT languages are substantially better.

On top of this, there is a new trend in which Artificial Intelligence-based approaches, such as [KLR<sup>+</sup>12, BCG19], “simply” learn the transformations themselves by inspecting a set of pairs of input and output models.

These two factors combined suggest a future where specialized MTLs may become irrelevant. We believe the transformation community needs to reflect on this scenario and discuss whether this can actually happen and, if so, whether something needs to be done about it.

With this purpose, we designed and distributed an online survey to have an insight of what academics and practitioners think about the current state and future of model transformation languages. We then presented the results of this survey during the 12th edition of the International Conference on Model Transformations (ICMT’19), followed by an open discussion on these topics not only during the conference itself, but also afterwards on social media. This paper reports the results of the survey and a summary of the opinions expressed afterwards. We hope this discussion sheds some light on the state of MTLs and its impact on us as researchers on model transformation topics.

The rest of the paper is structured as follows. Section 2 gives an overview of the current state of the art on model transformations. Section 3 presents the methodology we have followed to carry out the online survey and open discussion; Section 4 discusses its results. Finally, in Section 5, we conclude our work.

## 2 State of the Art

Model transformation is an instantiation of the more general problem of data transformation, which shows up across many fields of computer science, where the need to transform data either for format translation or refinement purposes often arises. As an example, this topic has been largely addressed by the (relational) database community, especially dealing with the heterogeneity of the data sources and the impedance mismatch problems [BM07, TBU10, BMR11].

In the modeling domain, this transformation problem was mainly addressed by first proposing a set of dedicated model transformation languages that designers could use to specify and execute a model transformation [CH06, MG06]. Followed by the improvements on these languages (and their associated tooling) so that they could to be used in practice.

In terms of performance, new incremental [EKK<sup>+</sup>13], parallel [BWV16], distributed [BGTC18] or NoSQL-based [DSC18] transformation engines were developed.

On the usability side, Model Transformation By-Example (MTBE) is an attempt to simplify the writing of exogenous model transformations [KLR<sup>+</sup>12, Var06, WSKK07, GGF09, BV09]. In an MTBE approach, users have to provide source models, their corresponding target models, as well as the correspondence between them (and for which a correspondence language has to be used). From this, the MTBE approach generates partial mappings that form the basis of the transformation.

Despite these advancements, we do not observe an increasing adoption and research on model transformation languages. In fact, some works like Kessentini et al. [KSBB12, MK14] and [BCG19] propose using search-based and neural network techniques to completely replace the writing of the transformation itself. And, based on informal chats with industrial people, even Java or other general-purpose languages are seen as better options. This perception was partially confirmed in this empirical evaluation [HSB<sup>+</sup>18].

The following sections describe the empirical validation we conducted to see if these “anecdotal” evidences are confirmed by the model transformation community at large and the community discussion that ensued as a result.

### 3 Methodology

We have conducted an empirical study to gather relevant data on the status of MTLs. We have divided the study in two phases: (1) an online survey, followed by (2) a live and open discussion during an ICMT plenary session which started with the presentation of the results obtained in the online survey.

In the following subsections, we provide a summary of the most relevant methodological aspects of both phases.

#### 3.1 Survey

The survey was conducted with the goal to have a broader understanding of the opinion of the community about the future of Model Transformations Languages beyond the usual and informal talks when small groups of people gather together.

Our aim is to be objective and not draw our own conclusions but report on their thoughts and answers to questions such as: “Do people think that MTLs are still useful? If not, what do you think will replace them? If yes, how do you think they can remain relevant?”.

##### 3.1.1 Survey design

The survey was anonymous and the participants were asked to answer the following questions based on their own experience:

Q1 *Single-choice question*: How do you mostly transform/manipulate models?

- ☐ I use a model transformation language (ATL, ETL, QVT, GTs, ...)
- ☐ I use a general purpose language (e.g. Java)
- ☐ I don't write the transformation. I use semi-automatic approaches (e.g. transformation-by-example)
- ☐ Others

Q2 *Free text question*: If you use a MT Language, indicate which one/s (and why you chose them)

Q3 *Free text question*: If you don't use any MT Language, can you give at least a couple of reasons why not? (any kind of reasons: related to the languages themselves, to the associated tooling, to the community around them, ...)

Q4 *Single-choice question*: Do you think MT Languages are more or less popular than 5 years ago?

- ☐ Less popular
- ☐ More popular
- ☐ Same as before

Q5 *Single-choice question*: How do you see the future of model transformation languages?

- ☐ They will disappear

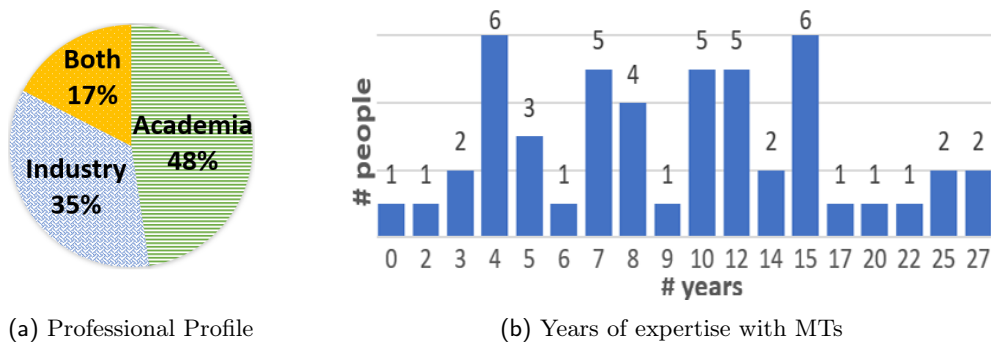


Figure 1 – Information about the participants

- ☐ They will become more popular
- ☐ They will survive but only in specific domains/applications
- ☐ Other scenarios

Q6 *Free text question:* Share any other thoughts you may have on the future of model transformation languages

### 3.1.2 Survey execution

The online survey was conducted in May 2019. We created an online form using SurveyMonkey<sup>1</sup>, and distributed it using our social networks, contacting people personally by email and sending invitations to distribution lists.

We recorded 63 responses. Among our interviewees, 48% claimed to work in academia, 35% in industry and 17% in both. They have been working with model transformation an average of 10.19 years with a standard deviation of 5.81 years. Figure 1 gives a graphical and more detailed view of this information.

We did not select the participants of the survey based on any criteria, so in theory they were randomly chosen, although given the community to which we belong—and thus, the people we are able to reach—we would say that the interviewees are somehow related to MDE.

The data analysis for the questions with quantitative or a finite number of answers—such as those with single-choice answers—was straight forward. The answers provided for open questions were manually studied and analyzed by ourselves. For Q2, we extracted the MTLs that each person mentioned. For Q3 and Q6, we identified the different reasons/thoughts provided and unified those that—although expressed in natural language in a different way—came to say the same. After this categorization process, we studied the frequency of these answers to detect recurrent answers.

## 3.2 Open discussion

The results of the online survey<sup>2</sup> were publicly presented during ICMT'19 and shared in a blog post<sup>3</sup>. After the presentation, we opened the floor to everyone who wanted to comment on the topic. There were 38 people in the room.

<sup>1</sup><https://www.surveymonkey.com>

<sup>2</sup>Slides: <https://www.slideshare.net/jcabot/is-there-a-future-for-model-transformation-languages>

<sup>3</sup><https://modeling-languages.com/future-model-transformation-languages/>

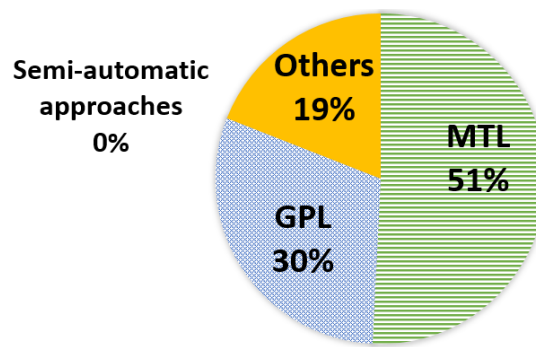


Figure 2 – Results for Q1: Use of Model Transformation Languages

Two different people took notes of everything said during the discussion and both versions were merged to obtain the final transcription.

A few additional opinions were versed after the ICMT discussion once some of the survey results were shared in social media. Being on social media, those comments had less context and clarifications than those shared during the face-to-face ICMT discussion but we were able to collect and add to the summary some additional comments, mostly from twitter.

## 4 Results

In this section, we first report the results obtained in the online survey and then present a summary of the discussions mainly generated during the ICMT plenary session in which they were presented.

### 4.1 Results of the Online Survey

Figure 2 shows that around a half of the respondents (51%) said they use domain-specific MTLs to write their model transformations, while 30% use general-purpose languages, and only 19% of the respondents transform their models using other means or they do not transform models at all. Interestingly, none of the respondents use (semi-)automatic approaches such as transformations-by-example, despite the number of research papers published on this topic.

Figure 3 presents the answer to question Q2 by means of a frequency histogram which shows the use of MTLs in practice. Only 49 people out of the 63 participants answered this question and they usually claimed to use more than one language.

Among the 20 different options we collected, clearly the most popular is ATL closely followed by QVTo (the imperative version of the QVT standard). The common reason the respondents gave for using ATL is its (apparent) simplicity, while people using QVTo said that they like it because of its maturity and the fact that it is supported by a standard. With respect to model-to-text transformation languages, Xtend is the most popular (as a modern GPL language) because of its precise syntax and the fact that no dedicated MT language is required. Beyond MTLs some respondents also mentioned GPLs they used: Java, C/C++ and the NodeJS runtime. Despite the existence of search-based approaches to MT, no one mentioned any of them.

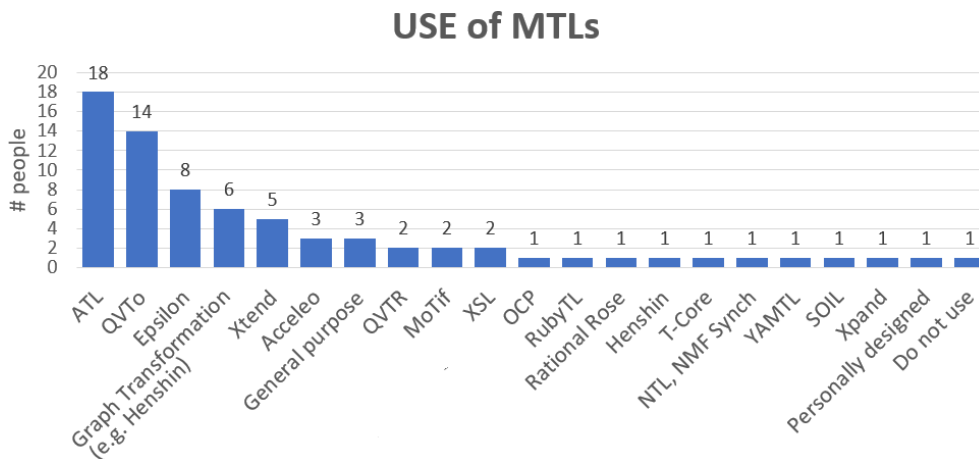


Figure 3 – Results for Q2: Model Transformation Languages Used in Practice

The analysis of the answers to Q3 led us to come up with the following main reasons (sorted by frequency) for not using MTLs:

They are complex to understand · They have a complex setup and IDE-specific dependencies · They have a high learning curve · The transformation engines behind the MTLs are mostly research prototypes · There is poor documentation · There is no real debugging · It is hard to hire staff · They are not adapted to the web · They do not have good performance · People without a background on MDE are skeptical · MTLs such as ATL are useful for simple transformations, but GPLs such as Java are needed for complex transformations.

Figure 4 presents the answers to questions Q4 and Q5. It shows that over half of the people think that MTLs have become less popular than 5 years ago, while only 10% believes that their adoption has increased. When asked about the future of MTLs, most of the interviewees (73%) think they will survive only on specific domains, 6% that they will disappear and only 1 person (2%) that they will become more popular. The remaining 20% did not make a prediction.

Finally, we summarized the answers to Q6 and divided them in two groups: one containing those in which people had positive/optimistic feelings and another in which people had negative/pessimistic feelings:

#### Positive

MTs are undoubtedly needed and useful · MTs are stupendous and they should be more used · MT may have longevity in data model transformations · No one will be able to design a system without models due to its complexity · MTL are important but breaking out from classical transformation problems and approaches is required

#### Negative

MTLs are not mature for industrial applications · Better showcase to industry is needed · There is lack of support from easily comparable, strong, empirical metrics · Steep learning curve · Not scalable · MTs are overrated and over-researched · Non-existence of a homogeneous community · Little standardization of useful languages · Dependency on proprietary tools · MTs suffer from the misconception of modeling

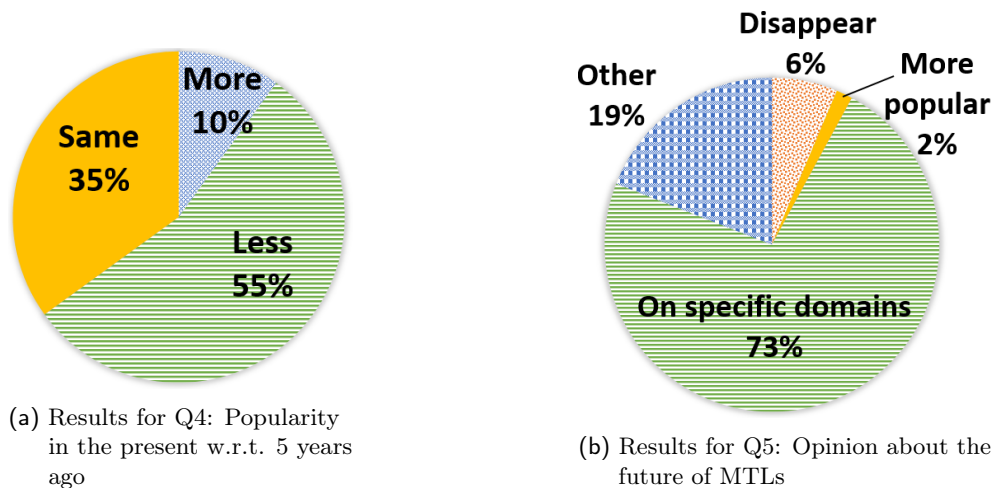


Figure 4 – Future

## 4.2 Summary of the Open Discussion

Immediately after presenting the survey results at ICMT'19, we opened the floor to discussion. Comments were varied, ranging from technical to social to business-oriented and everything in between. We present them in thematic groups.

Some people commented on the maturity and quality of MT tools, stating that no language has professional support (at least, this is the perception from industry) and that “often only a number between 2-10 people are involved in their development”. This low bus factor [CIC15] is a huge risk that many companies cannot assume.

While others agreed that this lack of industrialization of MTLs was an issue, they also pointed out that in the past a couple of companies have tried to provide a commercial offering on top of open source transformation tools but had later stopped due to lack of customer interest.

Beyond the tool industrialization, another participant also mentioned the need for a community around the tool that can help on the on-boarding of newcomers.

The attendees also commented that, years ago, the main selling point for MTLs was the emphasis on code generation for MDE, but in these days, this is no longer the case and it is model management that is now gaining importance. Some of them said that, in their opinion, doing model management (which involves model transformations) with a GPL might be problematic (e.g., code not maintainable, or many dependencies). Maybe this is an opportunity to refocus MTLs (and their features) to this challenge.

Although some people seemed to truly believe that there is a lot of potential in MT such as scalability, incrementality, analyses, bidirectionality, etc., some others criticized that this statement still lacks empirical evidence or perhaps simply that there are not that many companies that need such special functionality.

Students seem to be happy with MTLs (when they can make the tools work) so we should take this opportunity to convince them enough to make sure they keep favouring MTLs in the future. Nevertheless, there is also a risk that small student projects create false expectations. As one participant said: *I encountered that it was easy to do small examples in a MT language, but medium examples were unpleasant, and large examples awful – and nearly always better done in a GPL.*

Another interesting point raised in the discussion was that maybe MTLs had already made their valuable contribution to model-driven engineering. “MTL concepts are making their way into mainstream languages, e.g., newer Java and JVM-based languages, and graph databases such as Neo4j, which are powerful and work well”. From this perspective, MTLs are indeed dying as independent languages but they will live forever as part of other GPLs.

## 5 Conclusions

In this paper, we have presented the results of an online survey and a live and open discussion about the current state and future of model transformation languages.

Looking at the results of our study, we can conclude that there is an agreement on the fact that model transformation languages are becoming less popular but will remain being used in niches where their benefits can be more easily demonstrated. In this sense, probably MTLs are following the typical journey through the hype cycle<sup>4</sup>. After the “peak of inflated expectations” we are now climbing the “slope of enlightenment”.

There is also an agreement on the fact that negative results for MTLs are not (only) a technical issue but mostly due to social and tool aspects [BCJ<sup>+</sup>10] (i.e., knowledge and acceptance of MDE, lack of support and maintenance of MTLs, etc.) and due to the improvements in GPLs themselves that have integrated some of the ideas and programming constructs that years ago were only present in MTLs.

We can also conclude that new approaches such as search-based model transformations are not considered as an alternative to be used in practice for now. Probably because they are still mere research prototypes.

We believe any academic or practitioner interested in the field of model transformations can get some interesting insights from this work. Moreover, we also believe that this “exercise” was well appreciated by the community that felt it was important to have a collective discussion on these key topics. We hope this or similar discussions continue in the future including also quantitative evaluations. For instance, one of the suggestions from the open discussion was to replicate the experiments in [HSB<sup>+</sup>18] over alternative scenarios, e.g. one where traceability is important. Most MTLs are very good at keeping transformation traces and, therefore, these additional experiments could highlight use cases where MTLs are still clearly advantageous.

## References

- [BCG19] Loli Burgueño, Jordi Cabot, and Sébastien Gérard. An LSTM-based neural network architecture for model transformations. In *Proc. of the 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS’19)*, pages 294–299, 2019.
- [BCJ<sup>+</sup>10] Hugo Bruneliere, Jordi Cabot, Frédéric Jouault, Massimo Tisi, and Jean Bézivin. Industrialization of research tools: the ATL case. In *Proc. of the 3rd International Workshop on Academic Software Development Tools and Techniques @ ASE’2010*, September 2010.

---

<sup>4</sup>[https://en.wikipedia.org/wiki/Hype\\_cycle](https://en.wikipedia.org/wiki/Hype_cycle)



- [BGTC18] Amine Benelallam, Abel Gómez, Massimo Tisi, and Jordi Cabot. Distributing relational model transformation on mapreduce. *Journal of Systems and Software*, 142:1–20, 2018.
- [BM07] Philip A. Bernstein and Sergey Melnik. Model management 2.0: Manipulating richer mappings. In *Proc. of the ACM SIGMOD International Conference on Management of Data (SIGMOD’07)*, pages 1–12. ACM, 2007.
- [BMR11] Philip A. Bernstein, Jayant Madhavan, and Erhard Rahm. Generic schema matching, ten years later. volume 4, pages 695–701, 2011.
- [BV09] Zoltán Balogh and Dániel Varró. Model transformation by example using inductive logic programming. *Software and System Modeling*, 8(3):347–364, 2009.
- [BWV16] Loli Burgueño, Manuel Wimmer, and Antonio Vallecillo. A Linda-based platform for the parallel execution of out-place model transformations. *Information & Software Technology*, 79:17–35, 2016.
- [CH06] Krzysztof Czarnecki and Simon Helsen. Feature-based survey of model transformation approaches. *IBM Systems Journal*, 45(3):621–646, 2006.
- [CIC15] Valerio Cosentino, Javier Luis Cánovas Izquierdo, and Jordi Cabot. Assessing the bus factor of git repositories. In *Proc. of the 22nd IEEE International Conference on Software Analysis, Evolution, and Reengineering, (SANER’15)*, pages 499–503, 2015.
- [DSC18] Gwendal Daniel, Gerson Sunyé, and Jordi Cabot. Scalable queries and model transformations with the mogwai tool. In *Proc. of the 11th International Conference on Model Transformations (ICMT’18)*, pages 175–183, 2018.
- [EKK<sup>+</sup>13] Juergen Etzlstorfer, Angelika Kusel, Elisabeth Kapsammer, Philip Langer, Werner Retschitzegger, Johannes Schoenboeck, Wieland Schwinger, and Manuel Wimmer. A survey on incremental model transformation approaches. In *Proc. of the Workshop on Models and Evolution @ MODELS’13*, pages 4–13, 2013.
- [GGF09] Iván García-Magariño, Jorge J. Gómez-Sanz, and Rubén Fuentes-Fernández. Model transformation by-example: An algorithm for generating many-to-many transformation rules in several model transformation languages. In *Proc. of the 2nd International Conference on Model Transformations (ICMT’09)*, pages 52–66, 2009.
- [HSB<sup>+</sup>18] Regina Hebig, Christoph Seidl, Thorsten Berger, John Kook Pedersen, and Andrzej Wasowski. Model transformation languages under a magnifying glass: a controlled experiment with xtend, atl, and QVT. In *Proc. of the Conference and Symposium on the Foundations of Software Engineering (FSE’18)*, pages 445–455, 2018.
- [JABK08] Frédéric Jouault, Freddy Allilaire, Jean Bézivin, and Ivan Kurtev. ATL: A model transformation tool. *Science of Computer Programming*, 72(1-2):31–39, 2008.
- [KLR<sup>+</sup>12] Gerti Kappel, Philip Langer, Werner Retschitzegger, Wieland Schwinger, and Manuel Wimmer. Model transformation by-example: A survey of the first wave. In *Conceptual Modelling and Its Theoretical*

- Foundations: Essays Dedicated to Bernhard Thalheim on the Occasion of His 60th Birthday*, pages 197–215, 2012.
- [KPP08] Dimitrios S. Kolovos, Richard F. Paige, and Fiona A. C. Polack. The epsilon transformation language. In *Proc. of the International Conference on Model Transformations (ICMT'08)*, pages 46–60, 2008.
  - [KSBB12] Marouane Kessentini, Houari A. Sahraoui, Mounir Boukadoum, and Omar Benomar. Search-based model transformation by example. *Software and System Modeling*, 11(2):209–226, 2012.
  - [MG06] Tom Mens and Pieter Van Gorp. A taxonomy of model transformation. *Electr. Notes Theor. Comput. Sci.*, 152:125–142, 2006.
  - [MK14] Mohamed Wiem Mkaouer and Marouane Kessentini. Model transformation using multiobjective optimization. *Advances in Computers*, 92:161–202, 2014.
  - [OMG05] OMG. *MOF QVT Final Adopted Specification*. Object Management Group, 2005. OMG doc. ptc/05-11-01.
  - [SC12] Jesús Sánchez Cuadrado. Towards a family of model transformation languages. In *Proc. of the International Conference on Model Transformations (ICMT'12)*, pages 176–191, 2012.
  - [TBU10] James F. Terwilliger, Philip A. Bernstein, and Adi Unnithan. Automated co-evolution of conceptual models, physical databases, and mappings. In *Proc. of the International Conference on Conceptual Modeling (ER'10)*, pages 146–159, 2010.
  - [Var06] Dániel Varró. Model transformation by example. In *Proc. of the 9th International Conference on Model Driven Engineering Languages and Systems (MODELS'06)*, pages 410–424, 2006.
  - [WSKK07] Manuel Wimmer, Michael Strommer, Horst Kargl, and Gerhard Kramler. Towards model transformation generation by-example. In *Proc. of the 40th Hawaii International International Conference on Systems Science (HICSS'07)*, page 285, 2007.

## About the authors

**Loli Burgueño** is a postdoctoral researcher in the SOM Research Lab at the Internet Interdisciplinary Institute of the Open University of Catalonia (UOC) and CEA LIST. Contact her at [lbarguenoc@uoc.edu](mailto:lbarguenoc@uoc.edu), or visit <https://som-research.uoc.edu/loli-burgueno/>.

**Jordi Cabot** is an ICREA Research Professor at Internet Interdisciplinary Institute, the Research center of the Open University of Catalonia (UOC) where he is leading the SOM Research Lab. Contact him at [jordi.cabot@icrea.cat](mailto:jordi.cabot@icrea.cat), or visit <https://jordicabot.com/>.

**Sébastien Gérard** is research director at CEA and the research program leader around the knowledge co-engineering platform of the CEA LIST. Contact him at [sebastien.gerard@cea.fr](mailto:sebastien.gerard@cea.fr)

**Acknowledgments** We would like to thank Anthony Anjorin for his help taking notes during the open discussion. This work is supported by Spanish Research project TIN2016-75944-R, the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737494 and CEA in the context of the Modelia initiative.