In 2016...
6 panelists and 1 moderator...
They enjoyed their “arguments” so much ...
They wrote a paper about it ...
THIS IS THAT PAPER
User Experience for Model-Driven Engineering: Challenges and Future Directions
- a vision paper -

Silvia Abrahão ∗ Francis Bordeleau ∗ Betty HC Cheng
Sahar Kokaly ∗ Richard F. Paige ∗ Harald Störrle ∗ Jon Whittle

MODELS 2017
Austin, Texas, Sept 17-22, 2017
UX in MDE: State of the Practice
Scenario 1: Aerospace control software
Scenario 1: Aerospace control software

• Who is the user?
  • A team of software engineers (and possibly systems engineers)
• Need to support collaborative modeling capabilities
• Process
  • Requirements and constraints are captured in a non-MDE tool (DOORS, Word, etc.)
  • Customization of UML for the project at hand (UML profiles in tools such as Papyrus)
  • Models and constraints on models are produced iteratively
    • Constraints may be supplemented with “fixes”
  • A code generator is applied to produce code from models (e.g., Ada code)
  • Code generators may be extended to support new customizations or new domain constraints (e.g., timing analysis)
Scenario 1

• Question for interview:
  • With your systems engineer hat on – what challenges/difficulties do you see as a user of this approach?
• Points to bring up in answer:
  • How do I actually get started? How do I create a model, import a legacy model?
  • How do I connect models, and models with non-modelling artefacts? Where are these connections stored? How do they change when the models change?
• Question for interview:
  • System engineering is done within an organisation, working in a team – what showstoppers do you see for your team and organisation?
• Points to bring up in answer:
  • Collaboration: I have a team of 200 engineers, how do we work collaboratively on these artefacts?
  • It seems I have to switch frequently between views and applications (including between textual and graphical views). Can this be simplified?
  • How do I connect transformation tools (like ATL or Epsilon) to models in this workflow?
Scenario 2: Automotive control software

UML/SysML models + Stateflow/Simulink models
Scenario 2

• Question for interview:
  • This process is generic – as the MDE expert in your automotive software organisation, what tailoring do you carry out, and what difficulties do you encounter?
• Points for answer:
  • We build DSLs particularly to support simulation and to give domain experts a richer and more specific vocabulary.
  • DSLs are usually built with EMF/Ecore, and the EMF tooling introduces more affordances and mode confusion (e.g., why do I have to generate Ecore, why do I have to register my metamodel?)
• Question for interview:
  • What showstoppers are there for your automotive software organisation in using this MDE approach?
• Points for answer:
  • Exposing metamodels (Ecore/EMF) to authors of transformations and code generators: can this be avoided?
  • When models are updated, we have to re-run our transformations – this is inefficient. Can this be avoided?
Common MDE tasks performed by users

- Constructing Models (collaboratively, individually)
- Customizing modeling languages (via profiles, via annotations, via creating DSLs)
- Analysing models (verification, validation)
  - Testing models
  - Carrying out performance analysis
  - Model simulation
- Managing models (transformation, merging, comparison)
  - Defining model transformations
  - Run-time configuration and management of systems (models@runtime)
- Configuring modeling platforms
  - Common architectures, standards and repositories that supports the tasks above
The Systems Engineer

- How do I get started?
- How do I connect this model with other systems artefacts?
- How do I interface with my continuous integration server?
- How do I do versioning?
- How do my 200 colleagues work on this too?
How do I switch between graphical and textual views?

How do changes to my UML impact on my OCL?

Why do I have to switch views so frequently?

What is this textual representation of UML anyway?!
The Language Engineer

- What’s the difference between EMF and Ecore?
- Why do I have to generate Ecore?
- Do I need a tree-based editor?
- Should I use Emfatic, Java, UML, … to produce my metamodel?
- Why do I have to register my metamodel?
- ???!
The Model Transformer

- This switching back and forth between my model and transformation program is a pain!
- If I update my model, can I see what parts of the transformation program should change?
- I’m an expert in transformation but not metamodelling - can you shield me from metamodelling infrastructure?
The Engineer/Language Engineer

- How do I create, modify, extend my UML profile?
- How do I ensure that my model transformations are profile-aware?
- How do I generate all of this profile information?
- How do I use a UML profile along with unprofiled UML in the same project?
The Domain Expert

- How do my Simulink models integrate with the rest of my system models?
- How do I use my model transformations on my Simulink models?
- How do I bridge the MDE and non-MDE technical spaces???
User Experience: Brief Overview
UX vs. Usability

- Objective efficiency/effectiveness measures do not indicate how users feel.
- UX goes beyond traditional usability.
- UX shifts away from task-oriented interactions to focus on:
  - Hedonic qualities (e.g. is the product stimulating?), positive emotions, and affect (e.g. am I enthusiastic?)
User Experience in MDE

• These different stakeholders are having different user experiences.

  “The person’s emotion and attitudes about using a product, system or service. It includes both the practical, experiential, affective aspects of human–computer interaction (e.g., trust, engagements) and the person’s perceptions of system quality in use (e.g., ease of use, efficiency).”

• The MDE users are typically having negative user experiences:
  • Inconsistent and ineffective user interfaces.
  • Inefficiencies in “stitching together” different MDE tools.
  • Ineffectiveness in integrating MDE and non-MDE tools in a seamless fashion.
  • Unsatisfactory experiences in solving engineering problems.
User Experience in MDE

• These different stakeholders are having different user experiences.

“The person’s emotion and attitudes about using a product, system or service. It includes both the practical, experiential, affective aspects of human–computer interaction (e.g., trust, engagements) and the person’s perceptions of system quality in use (e.g., ease of use, efficiency)”.
Vision for UX in MDE

- **MX = User Experience for Model-Driven Engineering**
- **MX offers new angles to UX**
  - Focused on the multi-dimensional challenge of *integration*

---

**Model Integration**
Integration between MDE process and environment

**Tool Integration**

---

**Requirements Level**
- Requirements
- Scenarios
- Use Cases
- Feature Model

**System Level**
- SysML Model
- BPEL Model

**Design Level**
- UML Model
- Simulink Model
- State Model

**Supporting tools**

**Code generation:**
Automatic and/or manual

---

**Horizontal Integration**

**Vertical Integration (across abstraction levels)**
UX Process

Discover
- Observing
- Listening
- Learning

Evaluate
- Testing
- Measuring
- Improving

Design
- Reflecting
- Brainstorming
- Sketching
Most Frequent UX Methods

- **Test**
  - Requirements gathering: 84%
  - Field study / user interview: 62%
  - Diary study: 58%
  - Accessibility evaluation: 51%
  - Explore
  - Task analysis: 92%
  - Journey mapping: 83%
  - Design review: 82%
  - Write user stories: 80%
  - Clickable-prototype testing: 80%
  - Personas: 77%
  - Paper-prototype testing: 68%
  - Competitive analysis: 61%
  - Card sorting: 48%

Source: Nielsen Norman Group, Feb 12, 2017
https://www.nngroup.com/articles/ux-research-cheat-sheet/
UX in Software Engineering

Usability in SE

- Well recognized. Efforts on addressing usability for requirements gathering, software architecture design, component selection...
- Approaches for integrating usability in different software development processes (e.g., agile, MDE)

UX in SE

- It is still taking shape. Most initiatives deal with the integration of UX in agile methods.
- Maturity models for guiding companies with the integration of UX in agile methods start to appear.

Outside the SE/MDE communities

- Long history of considering the role of user. Efforts on how to take users’ existing practices into account when designing software tools.
- Considering social and organizational contexts
- Participatory-design and co-design
UX in MDE: Challenges and Future Directions
First, what are the goals of MDE?

From an industrial perspective, main goals for using MDE are to:

- Increase productivity (i.e., reduce development time and cost)
- Increase system quality
- Increase overall business and technical agility
Five UX Related Challenges for MDE:

User Model Integration  Processes for Tailoring UX  Empirical Studies

Customization and DSM Support  Interoperability
Challenge 1: User Model Integration

• Moving MDE from technology-driven approach → user-driven approach
• Broadening adoption - requires considering users with diverse backgrounds
• Key questions
  • Who are the users? e.g. developers, tool smiths, customers, suppliers
  • What are their main activities/concerns/needs?
  • What are the business goals, interests, motivations and values?
• UX design must focus on improving user productivity and satisfaction
  • Requires on-going UX measurement, monitoring and improvement

Future directions
• How can we integrate a “User Model” in the tool development process and make the user a first-class concern?
  • User background, domain and expertise
  • Tasks that different types of users are trying to accomplish
  • Work environment, which includes company policies, regulations, etc.

Make the user a first class concern
Challenge 2: Customization and DSM Support

• One size (MDE tool) does not fit all!
• Domain-specific tools UX > General purpose tools UX

Main challenges:
• Tool support for customization and DSML (to fit domain/context/type of users)
  • Support for tool simplification – adapt tool environment to each type of user
  • Support for workflow customization -- adapt tool workflows for specific dev context
  • Support for visual representations of domain concepts to different users
• Different DSMLs/tools are needed for different aspects of complex systems
  • To ensure overall consistency and enable system maintenance/evolution, models produced for different aspects must be integrated
  • One way to reduce complexity/cost of integration is to base DSMLs on the same underlying language (or meta-model), e.g. UML or EMF
    • Not possible/practical to base all aspects on same language, but should consolidate as much as possible
Customization and DSM Support - *Future Directions*

- UML tools have failed to properly support customization and DSML
  - This is not a language issue, but a tool issue!
- DSL tools attempt to solve this problem... but... beware of the “DSL-Babel”!
- “UML or DSL?” is simply a misleading question
  - Papyrus (UML) and Sirius (EMF) are two examples of tools supporting DSM
  - Right question: “What language do you want to base your DSL on?”

**Future directions**

- How can we improve tool support for customization and DSML?
- How can we support the composition of different DSMLs in the same development environment?
- How can we easily adapt capabilities like model diff/merge and search for specific DSMLs?

*The main issues/challenges are related to tool support, not languages!*
Challenge 3: Interoperability

- **Four** dimensions of interoperability:

**Model Integration**
- Both *vertical* (across abstraction levels) and *horizontal* (different views, different engineers, stakeholders, etc.)

**Tool Integration**
- E.g., integrating and MDE tool with a non MDE tool, combining different MDE tools through middleware or shared repositories

**Process Integration**
- E.g., combining support for model transformation and test-driven development through an IDE

**Integration through collaboration**
- E.g., support for integration among developers

Lack of interoperability causes context-switching overhead.
Interoperability - Future Directions

Support for Model and Tool Integration
- Increase in system complexity → more demands on modeling community
  - Development of new DSLs
  - New profiles on top of existing languages
  - New modeling concerns (e.g., run-time monitoring for autonomous systems)
- Need to maintain consistency throughout the development process and beyond (e.g., testing, run-time activities, etc.)
  - Both vertical and horizontal integrations need to be addressed
- Language integration and tool integration
  - Well defined languages, well defined integrations, traceability as languages/tools evolve
  - GEMOC and SLE both work in this direction

Support for Collaborative Modeling
- MDE tools lack (cf. programming IDEs) capabilities such as diff/merge and code reviews
  - Model diff/merge considered unreliable/hard to use, no real support for model reviews
- With no support for Collaborative Modeling → MDE tools difficult to use in industry

Consider all variants of interoperability and make it seamless.
Challenge 4: Processes for Tailoring UX

1: Modeling UX

What does UX mean for MDE?
- What is unit of analysis? (e.g., individual end-user interaction with tool or multiple user interactions?)
- Which dimensions (e.g., functionality, usability, value, pleasure, etc.) are relevant?
- What contextual factors (e.g., technology, physical environment, earlier experiences, task context, etc.) affect UX in MDE?

2: Design for UX

How can we consider UX within MDE scenarios?
- **Process**: How can we assess MDE tool (both digital and “pen and paper”) load factors on users? Some techniques: Participatory design, focus groups, design science, etc.
- **Product**: How can we improve UX for UML, Simulink, custom DSLs? Some techniques: Physics of Notation, Cognitive Dimension Framework, etc.

3: Evaluation and Improvement

When should UX be evaluated?
- e.g., during user interaction, before user interaction, after user interaction?

How to evaluate UX in early stages of language/tool development?
- What are methods to evaluate UX of anticipated use (i.e., without actual language/tool)?

How to operationalize and evaluate UX in MDE w.r.t measurements?
- What should be measured? (e.g., focus time, actions per task, etc.)
- What measurements results are specific to MDE vs. IDE/tool under evaluation?
Processes for Tailoring UX – Future Directions

A Theory for MX
- In mature sciences, empirical theories help gain and accumulate knowledge
- Need to build an explanatory framework with a number of factors that explain UX for MDE
- Existing theories from SE: e.g., theory for explaining the effect of UML-based development
- From Behavioral Sciences/Cognitive Psychology: theory of cognitive fit – adapt to MDE?

Engaging Other Disciplines
- E.g., cognitive behavioural scientists, graphical design, visualization technology experts, game designers, domain experts, etc.

Training and Support
- Cannot rely on books and passive/static documentation only
- Need a community-based approach (e.g., StackOverflow/YouTube videos)
- Learn from success stories (e.g., MathWorks); introduce modeling tools early in the academic process

Learn from the success of widely used modeling tools
Challenge 5: Empirical Studies of UX in MDE

• Some studies exist on *usability in MDE*
• Need a *body of knowledge* about MX (i.e., MX-BOK)
• Empirical research → theoretical advancements in MX
  • evaluate use of UX techniques proposed by HCI community in MDE context
  • understand how MDE users actually work (*positive, negative* experiences)

**Future directions:**

• Methods/metrics for evaluating MX with a sound theoretical basis
  • Consider methods/metrics from UX, adapting to MDE artefacts, concepts and tools
• Consider both *qualitative* and *quantitative* studies for MX

*Consider an *incremental* evaluation of MX (not a *post facto* evaluation)*
Takeaway Messages
Take-Away Message #1: 

*Manage Expectations*

The MDE research community has been successful in developing powerful tools used to solve significant industrial problems...

But...

Can we claim that our tools are *useful*?

- Cannot make claims about usability without providing empirical evaluation data.
- Claiming “Holistic” or “Systemic” support? Need usability evidence!
- Do not falsely advertise your MDE tool!
Take-Away Message #2:

*Empirically validate usability and improve*

Have an existing MDE Tool?

**Recommendation:** Evaluate its usability and use that process to trigger improvements (e.g., to interfaces).
Take-Away Message #3: *Consider usability as a first-class concern*

Working on a new MDE tool?
- Can you consider usability *up front* (i.e., before the tool or metamodel have been built)?
- What is the added value of doing so?
- Does it make sense to iteratively and incrementally evaluate usability while an MDE tool is being developed or applied?
In a nutshell...

Our work has identified:

- **five challenges** specific to achieving UX in MDE
- **future directions** to set the stage for a research program on UX in MDE

We hope this will benefit *tool builders* and *users* of MDE and the *software engineering community* as a whole!
User Experience for Model-Driven Engineering: Challenges and Future Directions - a vision paper -

Silvia Abrahão * Francis Bordeleau * Betty HC Cheng * Sahar Kokaly * Richard F. Paige * Harald Störrle * Jon Whittle

MODELS 2017
Austin, Texas, Sept 17-22, 2017