Towards Cross-Disciplinary Practices: Software Modeling for Enterprise, Business and other Domain Engineering Fields

Jean Bézivin
University of Nantes

JBezivin@gmail.com
JBezivin@twitter
1. The new engineering landscape
   1. Problem and Solution Spaces
   2. Domain Engineering
   3. Support Engineering
   4. Inter-Disciplinary and Trans-Disciplinary practices
   5. Conclusion

2. Software modeling as an enabling technology
   1. History of Software Modeling
   2. Why and how software modeling is important to engineering

3. Towards a next generation software modeling platform
   Santa Claus does not exist

1967
1980
1987
1998
2008

There is NO silver bullet
Problem and Solution Spaces

THE NEW ENGINEERING LANDSCAPE
Focus on Engineering

Scientists study the world as it is; engineers create the world that has never been.

Theodore von Kármán
What has changed in the past 50 years?

- Expressions like “CAD” or “Computer Assisted” or “Computer Aided” have lost all their discriminant meaning in engineering.

- Most engineering fields are now using computers and software.

- Time to adapt our vision.
Change of focus

Computers everywhere

Software everywhere

Models everywhere?

Computer Assisted Design

Software Intensive Systems

Model Driven Engineering?
The two engineering spaces

Problems lie here

Domain Engineering

Tools to solve problems may be found here

Support Engineering
Problems and Solutions

- Domain Engineering
  - Business engineering
  - Enterprise Engineering
  - Biological engineering
  - Automotive engineering
  - Health engineering
  - Civil engineering
  - Building engineering
  - Electrical engineering
  - Mechanical engineering

- Support Engineering
  - Process engineering
  - Product (line) engineering
  - Software language engineering
  - Model engineering
  - Service engineering
  - Data engineering
  - Program engineering
  - Event engineering
  - Constraint engineering
  - System engineering
  - Requirement engineering
  - Ontology engineering
  - OSS engineering
Problem Spaces

DOMAIN ENGINEERING
Traditional Engineering Fields:
Civil, Mechanical, Chemical, ...
Emerging Engineering Fields: Financial, Business, Enterprise, ...
Many features common to all domain engineering fields

- Based on support engineering
  - Products, Processes, Services, Objects, Rules, Processes, ...
  - Including HR and team management
    - Human in the loop
    - Engineers in control

- Chain
  1. Building Abstract Models
  2. Verification/Validation
  3. Putting in Production
  4. Putting in Operation

- Need for a strong model repository
  - Scaling up to millions of parts
  - Cooperative concurrent access
  - Point of view mechanisms
  - Strong zooming mechanisms
Electrical Engineering

Building abstract models → Validation Verification → Putting in Production → Augmenting, Changing the world
Construction Engineering

- Building abstract models
- Validation
- Verification
- Putting in Production
- Augmenting, Changing the world
Diversity of the Domain Engineering Landscape

- Civil Engineering
- Electrical Engineering
- Automotive Engineering
- Architecture Engineering
- Medical Engineering
- Chemical Engineering
- Biological Engineering
- Telephone Engineering
- Military Engineering
- Financial Engineering
- Business Engineering
- Enterprise Engineering
- Ecology Engineering
- Agricultural Engineering
- Communication Engineering
- Other Engineering Fields
Multiple Communities, Many Journals
And many more

http://www.govengr.com/

**Neural engineering** (also known as Neuroengineering) is a discipline within biomedical engineering that uses engineering techniques to understand, repair, replace, enhance, or otherwise exploit the properties of neural systems.

**Healthcare Engineering**
Biomedical engineering
Computer-aided medical engineering
Medical/disease modeling
Rehabilitation engineering
Healthcare energy systems engineering
Healthcare support service engineering
Emergency response engineering
Engineering issues in public health and epidemiology
Aging Engineering and aging (elderly patient service)
Healthcare engineering education
...

*Journal of Neural Engineering* to help scientists, clinicians and engineers to understand, replace, repair and enhance the nervous system.
Beyond Technical Spaces

SUPPORT ENGINEERING
Basic duality

Problem spaces

Solution spaces (Support Engineering)

Domain Engineering

Product Engineering

Process Engineering
Process engineering encompasses a vast range of industries, such as chemical, petrochemical, mineral processing, advanced material, food, pharmaceutical, biotechnological, and software industries.

See also Concurrent Engineering
Program Engineering

- Short name: “programming”
- Not to be confused with Language engineering
- Long tradition of excellence
- Noble and visible part of SE
- Very difficult
- Many iterations and branches
  - Structured Programming
  - OO Programming
  - Functional Programming
  - Etc.

Good definitions allow avoiding sterile, futile, and non productive discussions

«Mal nommer les choses, c'est ajouter au malheur du monde» Albert Camus
[To misname things is to add misery to the world]
Team and Product management

- Team Management Engineering
  - Software Team Management Engineering
  - Agile Methods

- Product Lifecycle Management (PLM)
  - Product Line Engineering (incl. variability)
  - Software Product Line Engineering
Diversity of the Support Engineering Landscape

- Language Engineering
- Program Engineering
- Ontology Engineering
- Model Engineering
- Web Engineering
- Service Engineering
- Transformation Engineering
- Rule Engineering
- Complex Event Engineering
- Data Engineering
- Process Engineering
- Product Engineering
- HR Engineering
- Team Engineering
- Software Engineering
- OSS Engineering
But also ...

Fields of various granularity, from coarse to fine
Inter-Disciplinary and Trans-Disciplinary Practices

NO ENGINEERING FIELD IS AN ISLAND
Specialized engineering fields

- Language Engineering
- Software Language Engineering
- Grammar Engineering
- Ontology Engineering
- Model Engineering
- XML Engineering
Composite Engineering Fields

- Software Engineering
  - Computer Engineering
  - Program Engineering
  - Language Engineering
  - Model Engineering
  - Method Engineering
  - Etc.

But also:
- OSS Engineering
- Document Engineering
- Requirement Engineering
- Formal Method Engineering
- Data Engineering
- User Interface Eng.
- Usability Engineering
- HR Engineering
- Education Engineering
- Team Mgmt Engineering
- Legal Engineering
Many Possible Useful Collaborations Between Support Eng.
Once in a great while, a great idea makes it across the boundary of one discipline to take root in another. The adoption of Christopher Alexander’s patterns by the software community is one such event.

Jim Coplien
Transfer of expertise between engineering fields

Architectural engineering

Software engineering
Strange Encounters: ME meets OSS

- The Normative period (1996-2004)
- The Open Source period (2004-2010)

Mission
The Eclipse Modeling Project will focus on the evolution and promotion of model-based development technologies within the Eclipse community. It will unite projects falling into this classification to bring holistic model-based development capabilities to Eclipse.
ME and 3D scanning/printing engineering

Real World

Model

Guaranteed 100% code free

Real World
Using ME for cross-disciplinary approaches

[SOFTWARE] MODEL ENGINEERING
Not all models are software models, but most of them are
Definition Framework

- **(Software) Model Engineering (ME)** promotes the systematic use of models, metamodels and model transformations to achieve industrial goals.
- **Model Driven Engineering (MDE)** is the application of ME principles and tools to any given engineering field.
- **Model Driven Software Engineering (MDSE)**
- **Model Driven (Software) Development (MDD)**
- **Model Driven Code Generation (MDCG)**
- **Model Driven Reverse Engineering (MDrevE)**
- **But also**
  - Model Driven Business Engineering (MDbizE)
  - Model Driven System Engineering (MDsysE)
  - Model Driven Data Engineering
  - Model Driven Web Engineering
  - Model Driven Requirement Engineering
  - Model Driven Civil Engineering
  - Model Driven Biological Engineering
  - etc.
Thesis

Model Engineering is a support engineering potentially useful in most other domain and support engineering fields, for example:

- Software Engineering
- Process Engineering
- Service Engineering
- Data Engineering
- Enterprise Engineering
- Business Engineering
- Mechanical Engineering
- Biological Engineering
ME is not only for code generation

Initially MDA was for just software engineering, but the scope was progressively extended to include:

- Software engineering
- Data engineering
- System engineering
- Business engineering
- Enterprise engineering
- Telecommunication engineering
- Building engineering
- Electrical engineering
- Mechanical engineering
- Automotive engineering
- Aeronautical engineering
- Biological engineering
- Automotive engineering
- Health engineering
- Financial engineering
- etc.

Broadening application spectrum (+ EDOC, etc.)
The Seven Bridges of Königsberg (Euler)

- The city of Königsberg included two large islands connected to each other and the mainland by seven bridges.
- The problem was to find a walk through the city that would cross each bridge once and only once.
- Leonhard Euler, in 1735, proved that the problem has no solution.
- First, Euler pointed out that the choice of route inside each land mass is irrelevant. The only important feature of a route is the sequence of bridges crossed.
- In this pre-computer period, Euler defined a language of nodes and edges (graphs) to describe a real world problem and to express a solution (Eulerian paths).
Model Engineering: the roots

When we wish to solve a problem on a computer we often need to construct within the computer a model of that aspect of the real or conceptual world to which the solution of the problem will be applied...

C.A.R. Hoare, “Record Handling”, in Programming Languages, F. Genuys (ed.), Villard-de-Lans (near Grenoble, France), September 1966
Model Engineering: illustration

Same visual notation, different context, different meaning (Thick red dotted lines for bicycle lanes)

The legend is the metamodel
The two orthogonal dimensions of MDE
The powerful calculus of Model Engineering

- Typical Definition:
  - ME is a typed polymorphic calculus on labelled graphs
  - $Mt [\text{MM}t] : Ma [\text{MM}a] \rightarrow Mb [\text{MM}b]$
Several periods for ME (iterations/ generations)

- Contemplative models
- Implicit metamodels
- Explicit metamodels
- Transformable models
- Transformation models
- Megamodels
- etc.
The long history of modeling languages

No global consolidated history of Modeling Languages

(3S) Modeling Languages
Progress in the History of Modeling Languages
Juha-Pekka Tolvanen (Metaedit+)
A View of 20th and 21st Century Software Engineering

Barry Boehm, ICSE2006, Shanghai

Where is the logical line of progress?
Paradigm/Artifact changes \{step = 15y.\}

1965
Procedural
Technology
Procedures, Pascal, C, ...
Procedural refinement

1980
Object
Technology
Objects, Classes, Smalltalk, C++, ...
Object composition

1995
Component
Technology
Components, Packages, Frameworks, Patterns, EJB, J2EE
Model transformation

2010
Model Driven
Engineering
Models, Metamodels, UML, MOF, ...

2025
Climbing the steps
Software engineering: Approaching half-time?

- 1965
- 1967
- 2015
- 2065

- Structured Programming
- Object Oriented Programming
- Agile Development
- ?

http://bertrandmeyer.com/2013/04/04/the-origin-of-software-engineering/
A possible scenario for MDE

ME is too important to be confined to pure software engineering
We learnt many things from the MDE last iteration

1. **Representation principle**
   - Any model $M$ represents a system $S$

2. **Multiple view principle**
   - A system $S$ may be represented by several models

3. **Conformance principle**
   - Any model $M$ conforms to the language of its metamodel $MM$

4. **3-level organization**
   - Any metamodel $MM$ conforms to a common metametamodel $MMM$ (3-level conjecture)

5. **Transformation principle**
   - The most important operation applicable to models is a transformation

6. **HOT principle**
   - A transformation is a model

7. **Weaving principle**
   - Abstract declarative correspondences between models may be represented as models

8. **Megamodel**
   - A model which elements are models or metadata and relations between these models

9. **Unification principle**
   - All models specialize a common abstract model

10. **Technical Space Framework**
    - Any model has a given representation defined by its technical space (no MOF/ECORE lock-in)
Example 1: Various accounting metamodels
(see also various business objects, process, rules,… metamodels)
Example 2: ME and BE:
The importance of declarative correspondence models

Business Model <-> Alignment Model <-> IT Model

Air France Business Model <-> Alignment Model <-> KLM Business Model
Software Engineering is Engineering

CONCLUSIONS
Model Engineering is no more part of Software Engineering only
The second life of Model Engineering

Most results achieved in the past iteration of ME (1995-2015) are applicable in this wider context.
ME and the semantic map of engineering fields

- Most projects have now become complex, interdisciplinary and interdependent, drawing on multiple engineering fields.
- We need a regular organization of all these support and domain engineering fields, showing the semantic relations between them.
  - For industry, education and research
- Model Engineering is one important piece in this new engineering landscape, allowing to:
  - Cope with various engineering fields
  - Use similar concepts and tools across all disciplines
  - Handle multidisciplinary projects
Thanks

• Questions?
• Comments?

JBezivin@gmail.com
JBezivin@twitter
Towards Cross-Disciplinary Practices: Software Modeling for Enterprise, Business and other Domain Engineering Fields

The world of engineering has changed a lot in the last fifty years. We realize that computers have become omnipresent and software is ubiquitous. Moreover most classical and emerging domain engineering fields now heavily draw on some forms of software modeling techniques and tools. Time is coming to look for some “unifying theory of engineering” and imagine possible corresponding conceptual frameworks and even generic platforms. To make things concrete, we can consider two broad categories of engineering fields called “support engineering” on one side and “domain engineering” on the other side. The first category defines a set of technical spaces like service engineering, system engineering, model engineering, constraint engineering, data engineering, process engineering, event engineering, language engineering, program engineering, formal methods engineering, and many more. At the opposite of this solution space, we find the problem space with a lot of conventional or emerging domain engineering fields like electrical, mechanical, civil, telecommunication, automotive, avionics, health, biological, financial, business, enterprise and many more. There are several commonalities between domain engineering that would gain to be exposed: starting with the construction of abstract models conforming to some ontology, a second step usually defines some model validation or verification followed by a manufacturing or production step and finally a deployment step intended to augment or transform the real world. The presentation will propose an initial cartography of support and domain engineering, illustrating its possible impact on the organization of research and advanced education. It will also emphasize the important place taken by software model engineering in this possible organization, with examples from several fields including enterprise and business engineering.