THE MODEL-BASED SOFTWARE ENGINEERING BODY OF KNOWLEDGE (MBEBOK)

OOPSLE 2020

Alfonso Pierantonio

Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Università degli Studi dell'Aquila

Joint work with

L. Burgueño, F. Ciccozzi, M. Famelis, G. Kappel, L. Lambers, S. Mosser, R.F. Paige, A. Rensink, R. Salay, G. Taentzer, A. Vallecillo, M. Wimmer

Great things are done when men and mountains meet

W. Blake

1st Winter Modeling Meeting

San Vigilio di Marebbe, Italy – January 2018

1st Winter Modeling Meeting

San Vigilio di Marebbe, Italy – January 2018

Model-Based Software Engineering (MBE) is a widely accepted Software Engineering (SE) discipline

Model-Based Software Engineering (MBE) is a widely accepted Software Engineering (SE) discipline **however no agreement upon core set of concepts and practices has been met for it yet**

A Body of Knowledge should help define concepts, terms and activities that make up Model-Based Software Engineering

ROADMAP

- What is a Body of Knowledge
- Related Bodies of Knowledge
  - The Software Engineering BoK (SWEBOK)
  - The Software Language Engineering BoK (SLEBOK)
- The Model-Based Software Engineering BoK (MSEBOK)
  - Topics
  - Survey
- Proposal

BODY OF KNOWLEDGE

A Body of Knowledge is a set of concepts, terminology, and tasks that constitute a professional domain

A BoK is the accepted ontology for a specific domain

- a representation, formal naming and definition of the categories, properties and relations between the concepts, data and entities that substantiate one, many or all domains of discourse
- A BoK is developed by professional associations to capture inherent knowledge that occurs in a professional domain

BODY OF KNOWLEDGE

Set of concepts, terminology, and tasks that constitute a professional domain

The main goals of a BoK on a given discipline are

- to promote a consistent and global view of the discipline
- to specify the scope of the discipline
- to clarify its place with respect to other related disciplines
- to characterize the contents, and known practices of the discipline, organizing them in a coherent and comprehensive manner;
- to provide a foundation for curriculum development and, when applicable, for individual certification and licensing material

BODY OF KNOWLEDGE

Set of concepts, terminology, and tasks that constitute a professional domain

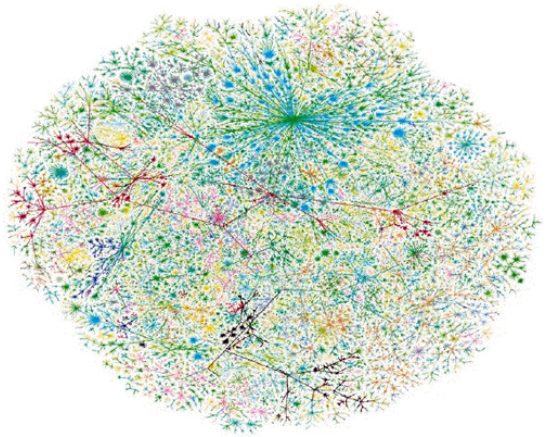
A BoK should also provide concrete deliverables

BODY OF KNOWLEDGE

Set of concepts, terminology, and tasks that constitute a professional domain

A BoK should also provide concrete deliverables

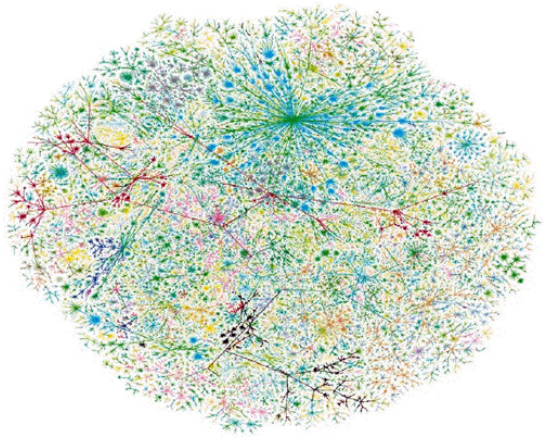
- A terminology that defines the set of main concepts of the discipline, as used by their practitioners: an accepted ontology for the specific domain.



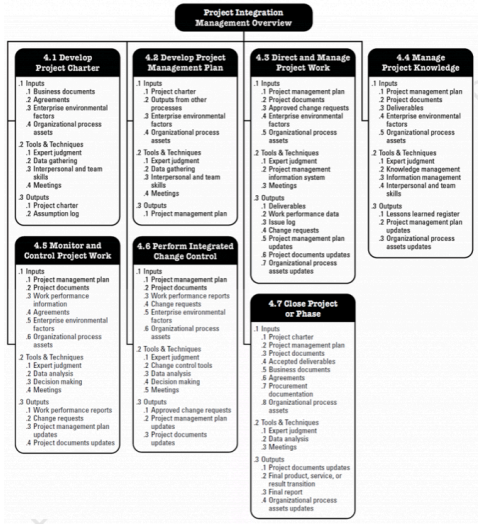
BODY OF KNOWLEDGE

Set of concepts, terminology, and tasks that constitute a professional domain

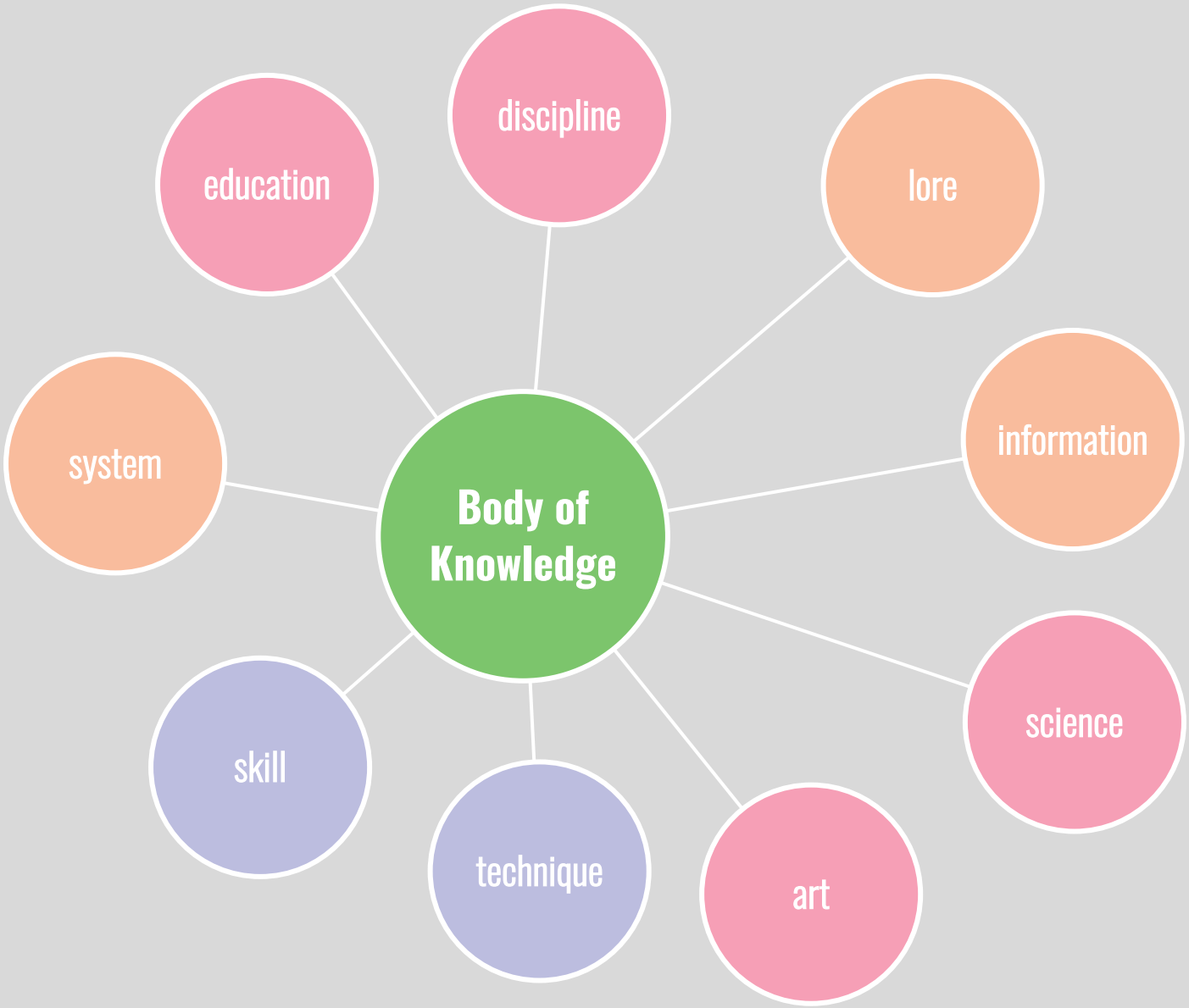
A BoK should also provide concrete deliverables



- A terminology that defines the set of main concepts of the discipline, as used by their practitioners: an accepted ontology for the specific domain.



- A structured list of the main knowledge areas, skills and accepted practices of the discipline, covering all the basic knowledge that any practitioner should possess.


Body of Knowledge
discipline
education
lore
system
information
skill
science
technique
art

BODY OF KNOWLEDGE

A BoK should always be descriptive, but not prescriptive: it should not impose any particular method or tool, nor specific practices

- Generally recognized

  the knowledge and practices described are generally applicable to multiple kinds of diversified projects in various situations, and there is consensus about their value and usefulness.

- Good practices

  there is general agreement that the application of skills, tools, and techniques can enhance the chances of success over a wide range of projects.

BODY OF KNOWLEDGE

Existing Bodies of Knowledge

A number of BoKs for various software-related disciplines :

- Software Engineering BoK (SWEBOK)
- Software Language Engineering BoK (SLEBOK)

- Enterprise Architecture BoK (EABOK)
- Business Analysis BoK (BABOK)
- Systems Engineering BoK (SEBOK)
- Data Management BoK (DMBOK)
- Project Management BoK (PMBOK)
- Automation BoK (ABOK)

RELATED BODIES OF KNOWLEDGE

THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE

- In 2004, a joint effort ACM/IEEE Computer society established the baseline for the BoK of the field of software engineering
- The mission was:

  to establish the appropriate set(s) of criteria and norms for professional practice of software engineering upon which industrial decisions, professional certification, and educational curricula can be based.

- The SWEBOK has been adopted by ISO and IEC as ISO/IEC TR 19759:2005.2

THE SOFTWARE ENGINEERING BODY OF KNOWLEDGE

How it has been developed

The SWEBOK was developed as an international collective effort, in order to provide a consistent global view of software engineering

- The committee appointed two chief editors, several co-editors to support them, and editors for each of the Knowledge Areas
- All chapters were openly reviewed, in an editing process that engaged approximately 150 reviewers from 33 countries
- Professional and scientific societies, as well as public agencies from all over the world involved in software engineering, were contacted, made aware of this project, and invited to participate in the review process too.

KNOWLEDGE AREAS IN SWEBOK

Software Engineering Models and Methods

One complete Knowledge Area is devoted to Software Engineering Models and Methods

- The scope was to
  - provide models and methods to impose structure on software engineering with the goal of making that activity systematic, repeatable, and ultimately more success-oriented
  - use models to provide an approach to problem solving, a notation, and procedures for model construction and analysis.
  - use methods to provide an approach to the systematic specification, design, construction, test, and verification of the end-item software and associated work products.

Modeling

- Modeling principles
- Properties and expressions of models
- Syntax, semantics, and pragmatics
- Preconditions, postconditions, and invariants

KNOWLEDGE AREAS IN SWEBOK

Software Engineering Models and Methods

Modeling

- Modeling principles
- Properties and expressions of models
- Syntax, semantics, and pragmatics
- Preconditions, postconditions, and invariants

Types of models

- Information modeling
- Behavioral modeling
- Structural modeling

KNOWLEDGE AREAS IN SWEBOK

Software Engineering Models and Methods

Modeling

- Modeling principles
- Properties and expressions of models
- Syntax, semantics, and pragmatics
- Preconditions, postconditions, and invariants

Types of models

- Information modeling
- Behavioral modeling
- Structural modeling

Analysis of models

- Analyzing for completeness
- Analyzing for consistency
- Analyzing for correctness
- Traceability
- Interaction Analysis

KNOWLEDGE AREAS IN SWEBOK

Software Engineering Models and Methods

Modeling

- Modeling principles
- Properties and expressions of models
- Syntax, semantics, and pragmatics
- Preconditions, postconditions, and invariants

Types of models

- Information modeling
- Behavioral modeling
- Structural modeling

Analysis of models

- Analyzing for completeness
- Analyzing for consistency
- Analyzing for correctness
- Traceability
- Interaction Analysis

Software Engineering Methods

- Heuristic methods
- Formal methods
- Prototyping methods
- Agile methods

RELATED BODIES OF KNOWLEDGE

- The Software Engineering BoK (SWEBOK)
- The Software Language Engineering BoK (SLEBOK)

THE SOFTWARE LANGUAGE BODY OF KNOWLEDGE

It is a **community-wide** effort that has been initiated in the SLEBoK Dagstuhl Seminar in 2017

- It features artefacts, definitions, methods, techniques, best practices, open challenges, case studies, teaching material, and other components
- It is intended for students, researchers, teachers, and practitioners
- In order to learn from, to better leverage, to better contribute to, and to better disseminate the intellectual contributions and practical tools and techniques coming from the SLE field

Anyone can contribute to the revision process via its Git repository

- http://slebok.github.io/

THE SOFTWARE LANGUAGE BODY OF KNOWLEDGE

The mission is providing a unique and comprehensive description of the concepts, tools and methods developed by the SLE community

- terms: provides a glossary and precise definitions of the concepts used and developped in SLE
- topics: connects terms into broader categories to form a proper taxonomy and eventually ontology
- literature: provides several reviews of the litterature on specific topics related to SLE
- tools: provides the description of some SLE tools
- usecases: describes relevant scenarios for SLE
- teaching: provides materials about existing courses, and insights regarding an SLE curriculum

The SLEBOK

- defines notions of model and metamodel which are not incompatible with MBEBOK, though MBEBOK's definitions are more elaborated
- does mention the notion of static semantics, but does not elaborate on language semantics, whereas MBEBOK explicitly captures model semantics as a key concept

Moreover

- SLEBOK and MBEBOK treat abstract and **concrete syntax** differently: while these are first-class concepts in MBEBOK, their notions are distributed across multiple concepts in SLEBOK

THE MODEL-BASED SOFTWARE ENGINEERING BODY OF KNOWLEDGE

TOPICS FOR AN MBEBOK

The MBEBOK has been initiated during the WMM 2018, its **list of topics** has been developed in two stages

1. An initial proposal was presented at the MODELS Educators' Symposium in October 2018 and discussed during the event

   A poster with the proposal was also prepared and displayed during the conference, and the participants were asked to comment

2. A survey study has been conducted about three main aspects:
   - the topics included in the list,
   - the relevance assigned to each of them, and
   - the coverage of the topics in the courses at both Bachelor and Master levels

TOPICS FOR AN MBEBOK

The MBEBOK has been initiated during the WMM 2018, its **list of topics** has been developed in two stages

1. An initial proposal was presented at the MODELS Educators' Symposium in October 2018 and discussed during the event

   A poster with the proposal was also prepared and displayed during the conference, and the participants were asked to comment

2. A survey study has been conducted about three main aspects:
   - the topics included in the list,
   - the relevance assigned to each of them, and
   - the coverage of the topics in the courses at both Bachelor and Master levels

TOPICS FOR AN MBEBOK

Publications

2018 **How do we teach modelling and model-driven engineering? A survey**. In Educators Symposium 2018 EduSymp18, MODELS, 16 Oct 2018, Copenhagen, Denmark.

2018 **Towards a body of knowledge for model-based software engineering** In Educators Symposium 2018 EduSymp18, MODELS, 16 Oct 2018, Copenhagen, Denmark

2019 **Contents for a Model-Based Software Engineering Body of Knowledge**. Journal of Software and System Modeling (2019) 18: 3193

TOPICS 1/3

relevant topics part of the knowledge that any practitioner should possess

1. Model Foundations

   basic modeling concepts, principles and practices

2. Model Quality

   quality aspects of models, including completeness, consistency, correctness, comprehensibility, confinement (i.e., fitness for purpose) and changeability

3. Analysis

   structural model analysis (consistency checking, instance generation, metrics and bad smell detection), behavioral model analysis (pre-/postcondition checking, simulation, performance analysis, reachability analysis, temporal model checking), and model transformation analysis (correctness, termination, etc.).

4. Modeling Languages

   language definition (metamodels, grammars, semantics), types of modeling languages (general purpose, domain-specific), and multiview modeling (model viewpoints and views, correspondences among views, viewpoint consistency, and viewpoint integration)

TOPICS 1/3

relevant topics part of the knowledge that any practitioner should possess

5. Model Representation
   concrete syntax, the physics of notations, layouts, the dichotomy between textual and visual models, as well as animation

6. Model Maintenance and Evolution
   model operations (diff, merge, refactoring), model versioning, and model migration

7. Model Execution
   model simulation and co-simulation, execution strategies (sequential vs. parallel), and model debugging and testing.

8. Model Transformations
   model transformation languages (syntax and semantics), model transformation types, and model application syntax

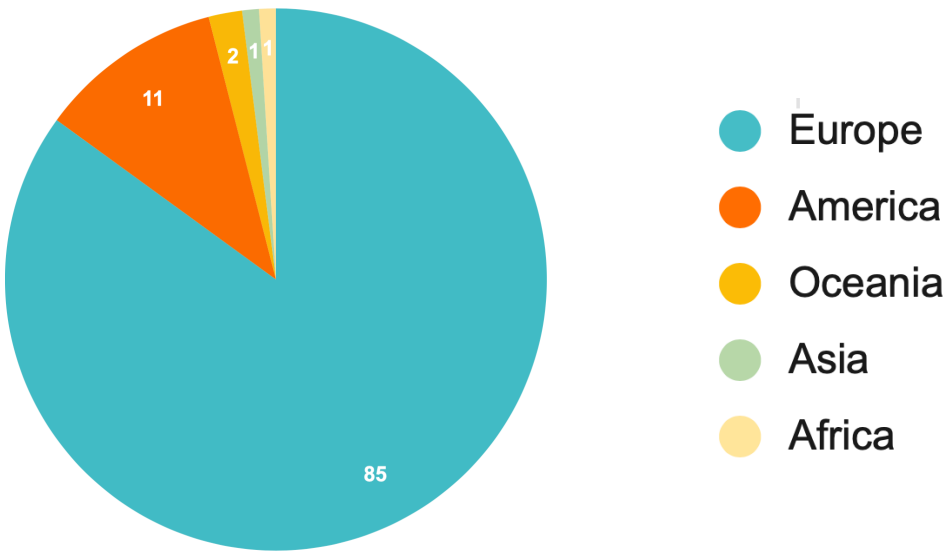
TOPICS 1/3

relevant topics part of the knowledge that any practitioner should possess

9. Further topics

   how MBE is used in application domains, advanced topics, application scenarios, engineering best practices

- The survey was conducted in December 2018
- An invitation to participate was sent to all major Software Engineering and Modeling distribution lists
  - https://encuestas.uma.es/27511/lang-en
- A total of 101 responses were recorded, from 23 different countries



A 4-point Likert scale was used to record the importance assigned to each topic, for example

*
1. Model Foundations: Importance

	Not important at all	Slightly important	Moderately important	Important	No opinion
1.1 Syntax	○	○	○	○	○
1.1.1 Abstract syntax. Metamodels	○	○	○	○	○
1.1.2 Concrete syntax	○	○	○	○	○
1.2 Semantics	○	○	○	○	○
1.2.1 Structural	○	○	○	○	○
1.2.2 Behavioral (discrete vs. continuous)	○	○	○	○	○
1.3 Purpose/intent	○	○	○	○	○
1.3.1 Modeling principles: abstraction, purpose, perspective, rigor, etc.	○	○	○	○	○
1.3.2 Exemplar purposes: such as Metamodeling or model transformation definition	○	○	○	○	○

SURVEY / SCORES

	Importance		BSc Coverage		MSc Coverage		Importance minus:	
	Avg	Std.dev	Avg	Std.dev	Avg	Std.dev	BSc cov	MSc cov
1. Model Foundations								
1.1. Syntax	3.73	*0.61*	2.53	*0.94*	3.04	*1.04*	1.21	0.69
* Abstract syntax. Metamodels	3.71	*0.63*	2.20	*1.00*	3.02	*1.07*	1.51	0.69
* Concrete syntax	3.57	*0.69*	2.56	*0.96*	2.93	*1.07*	1.02	0.65
1.2. Semantics	3.72	*0.60*	2.26	*0.91*	2.82	*1.07*	1.46	0.90
* Structural	3.66	*0.69*	2.33	*0.93*	2.77	*1.10*	1.34	0.89
* Behavioral (discrete vs. continuous)	3.54	*0.76*	2.05	*0.86*	2.48	*1.13*	1.49	1.06
1.3. Purpose/intent	3.73	*0.54*	2.28	*0.92*	2.87	*1.04*	1.45	0.86
* Modeling principles	3.76	*0.55*	2.37	*0.96*	2.88	*1.03*	1.39	0.88
* Exemplar purposes: such as Metamodeling or model transformation definition	3.41	*0.79*	1.85	*0.93*	2.70	*1.06*	1.55	0.70

- average score and standard deviation of the assigned importance and coverage at BSc and MSc levels. The last two columns show the differ- ence between the importance assigned to a topic and how it is covered in the curricula of the institution, in order to iden- tify possible decompensations. Cells shaded in green color highlight the highest scores, while red-shaded cells identify the lowest. The highest standard deviations are highlighted in yellow; they identify the topics with less consensus.

SURVEY / SCORES

	Importance		BSc Coverage		MSc Coverage		Importance minus:	
	Avg	Std.dev	Avg	Std.dev	Avg	Std.dev	BSc cov	MSc cov
1. Model Foundations								
1.1. Syntax	3.73	*0.61*	2.53	*0.94*	3.04	*1.04*	1.21	0.69
* Abstract syntax. Metamodels	3.71	*0.63*	2.20	*1.00*	3.02	*1.07*	1.51	0.69
* Concrete syntax	3.57	*0.69*	2.56	*0.96*	2.93	*1.07*	1.02	0.65
1.2. Semantics	3.72	*0.60*	2.26	*0.91*	2.82	*1.07*	1.46	0.90
* Structural	3.66	*0.69*	2.33	*0.93*	2.77	*1.10*	1.34	0.89
* Behavioral (discrete vs. continuous)	3.54	*0.76*	2.05	*0.86*	2.48	*1.13*	1.49	1.06
1.3. Purpose/intent	3.73	*0.54*	2.28	*0.92*	2.87	*1.04*	1.45	0.86
* Modeling principles	3.76	*0.55*	2.37	*0.96*	2.88	*1.03*	1.39	0.88
* Exemplar purposes: such as Metamodeling or model transformation definition	3.41	*0.79*	1.85	*0.93*	2.70	*1.06*	1.55	0.70

average score and standard deviation of the assigned importance

SURVEY / SCORES

	Importance		BSc Coverage		MSc Coverage		Importance minus:	
	Avg	Std.dev	Avg	Std.dev	Avg	Std.dev	BSc cov	MSc cov
1. Model Foundations								
1.1. Syntax	3.73	*0.61*	2.53	*0.94*	3.04	*1.04*	1.21	0.69
* Abstract syntax. Metamodels	3.71	*0.63*	2.20	*1.00*	3.02	*1.07*	1.51	0.69
* Concrete syntax	3.57	*0.69*	2.56	*0.96*	2.93	*1.07*	1.02	0.65
1.2. Semantics	3.72	*0.60*	2.26	*0.91*	2.82	*1.07*	1.46	0.90
* Structural	3.66	*0.69*	2.33	*0.93*	2.77	*1.10*	1.34	0.89
* Behavioral (discrete vs. continuous)	3.54	*0.76*	2.05	*0.86*	2.48	*1.13*	1.49	1.06
1.3. Purpose/intent	3.73	*0.54*	2.28	*0.92*	2.87	*1.04*	1.45	0.86
* Modeling principles	3.76	*0.55*	2.37	*0.96*	2.88	*1.03*	1.39	0.88
* Exemplar purposes: such as Metamodeling or model transformation definition	3.41	*0.79*	1.85	*0.93*	2.70	*1.06*	1.55	0.70

coverage at BSc and MSc levels

SURVEY / SCORES

	Importance		BSc Coverage		MSc Coverage		Importance minus:	
	Avg	Std.dev	Avg	Std.dev	Avg	Std.dev	BSc cov	MSc cov
1. Model Foundations								
1.1. Syntax	3.73	*0.61*	2.53	*0.94*	3.04	*1.04*	1.21	0.69
* Abstract syntax. Metamodels	3.71	*0.63*	2.20	*1.00*	3.02	*1.07*	1.51	0.69
* Concrete syntax	3.57	*0.69*	2.56	*0.96*	2.93	*1.07*	1.02	0.65
1.2. Semantics	3.72	*0.60*	2.26	*0.91*	2.82	*1.07*	1.46	0.90
* Structural	3.66	*0.69*	2.33	*0.93*	2.77	*1.10*	1.34	0.89
* Behavioral (discrete vs. continuous)	3.54	*0.76*	2.05	*0.86*	2.48	*1.13*	1.49	1.06
1.3. Purpose/intent	3.73	*0.54*	2.28	*0.92*	2.87	*1.04*	1.45	0.86
* Modeling principles	3.76	*0.55*	2.37	*0.96*	2.88	*1.03*	1.39	0.88
* Exemplar purposes: such as Metamodeling or model transformation definition	3.41	*0.79*	1.85	*0.93*	2.70	*1.06*	1.55	0.70

difference between the importance assigned to a topic and how it is covered in the curricula

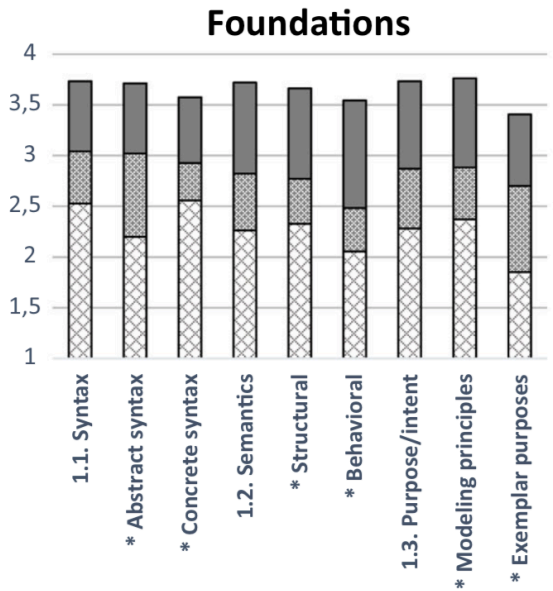
SURVEY / SCORES

Model Foundations

	Importance		BSc Coverage		MSc Coverage		Importance minus:	
	Avg	Std.dev	Avg	Std.dev	Avg	Std.dev	BSc cov	MSc cov
1. Model Foundations								
1.1. Syntax	3.73	*0.61*	2.53	*0.94*	3.04	*1.04*	1.21	0.69
* Abstract syntax. Metamodels	3.71	*0.63*	2.20	*1.00*	3.02	*1.07*	1.51	0.69
* Concrete syntax	3.57	*0.69*	2.56	*0.96*	2.93	*1.07*	1.02	0.65
1.2. Semantics	3.72	*0.60*	2.26	*0.91*	2.82	*1.07*	1.46	0.90
* Structural	3.66	*0.69*	2.33	*0.93*	2.77	*1.10*	1.34	0.89
* Behavioral (discrete vs. continuous)	3.54	*0.76*	2.05	*0.86*	2.48	*1.13*	1.49	1.06
1.3. Purpose/intent	3.73	*0.54*	2.28	*0.92*	2.87	*1.04*	1.45	0.86
* Modeling principles	3.76	*0.55*	2.37	*0.96*	2.88	*1.03*	1.39	0.88
* Exemplar purposes: such as Metamodeling or model transformation definition	3.41	*0.79*	1.85	*0.93*	2.70	*1.06*	1.55	0.70

SURVEY / SCORES
Model Foundations

	Importance		BSc Coverage		MSc Coverage		Importance minus:	
	Avg	Std.dev	Avg	Std.dev	Avg	Std.dev	BSc cov	MSc cov
1. Model Foundations								
1.1. Syntax	3.73	*0.61*	2.53	*0.94*	3.04	*1.04*	1.21	0.69
* Abstract syntax. Metamodels	3.71	*0.63*	2.20	*1.00*	3.02	*1.07*	1.51	0.69
* Concrete syntax	3.57	*0.69*	2.56	*0.96*	2.93	*1.07*	1.02	0.65
1.2. Semantics	3.72	*0.60*	2.26	*0.91*	2.82	*1.07*	1.46	0.90
* Structural	3.66	*0.69*	2.33	*0.93*	2.77	*1.10*	1.34	0.89
* Behavioral (discrete vs. continuous)	3.54	*0.76*	2.05	*0.86*	2.48	*1.13*	1.49	1.06
1.3. Purpose/intent	3.73	*0.54*	2.28	*0.92*	2.87	*1.04*	1.45	0.86
* Modeling principles	3.76	*0.55*	2.37	*0.96*	2.88	*1.03*	1.39	0.88
* Exemplar purposes: such as Metamodeling or model transformation definition	3.41	*0.79*	1.85	*0.93*	2.70	*1.06*	1.55	0.70



SURVEY / SCORES

Model Quality, Analysis

	Importance		BSc Coverage		MSc Coverage		Importance minus:	
	Avg	Std.dev	Avg	Std.dev	Avg	Std.dev	BSc cov	MSc cov
2. Model Quality								
2.1. Completeness	3.24	*0.85*	2.05	*0.92*	2.60	*1.07*	1.19	0.64
2.2. Consistency	3.77	*0.55*	2.19	*0.91*	2.82	*1.12*	1.58	0.95
2.3. Correctness	3.64	*0.59*	2.30	*0.97*	2.74	*1.13*	1.35	0.90
2.4. Comprehensibility	3.66	*0.57*	2.15	*0.98*	2.56	*1.09*	1.52	1.10
2.5. Confinement (= fitness for purpose)	3.57	*0.64*	1.98	*0.83*	2.48	*1.07*	1.59	1.09
2.6. Changeability	3.22	*0.83*	1.65	*0.77*	2.26	*1.07*	1.57	0.96
3. Analysis								
3.1. Structural model analysis	3.46	*0.71*	1.94	*0.93*	2.41	*1.04*	1.52	1.05
* Invariant checking	3.24	*0.89*	1.83	*0.93*	2.42	*1.09*	1.42	0.82
* Instance generation	3.22	*0.92*	1.72	*0.90*	2.22	*1.10*	1.51	1.00
* Metrics calculation	2.90	*0.83*	1.50	*0.71*	2.03	*1.00*	1.40	0.87
* Smells detection	2.91	*0.89*	1.49	*0.74*	1.89	*0.98*	1.41	1.02
3.2. Behavioral model analysis	3.37	*0.77*	1.69	*0.75*	2.22	*0.98*	1.68	1.15
* Pre-postcondition checking	3.28	*0.87*	1.68	*0.81*	2.21	*1.06*	1.60	1.07
* Simulation	3.29	*0.82*	1.57	*0.74*	2.13	*1.05*	1.73	1.16
* Reachability analysis	2.96	*0.86*	1.43	*0.71*	1.96	*1.05*	1.52	1.00
* *Temporal model checking*	*2.77*	*0.97*	*1.35*	*0.58*	*2.07*	*1.08*	*1.42*	*0.70*
* Performance	2.96	*0.95*	1.38	*0.61*	1.97	*1.03*	1.58	0.99
3.3. Model transformation analysis	3.24	*0.94*	1.41	*0.71*	2.15	*1.05*	1.83	1.09
* MT Correctness (of transformed models, in syntax and semantics)	3.30	*0.88*	1.35	*0.67*	2.10	*1.02*	1.95	1.20
* MT Completeness	3.00	*0.94*	1.32	*0.61*	1.98	*1.09*	1.68	1.02
* MT Functional behavior (termination, confluence)	2.93	*0.90*	1.30	*0.58*	1.99	*1.04*	1.63	0.94
* *MT Performance*	*2.78*	*0.93*	*1.30*	*0.58*	*1.86*	*1.05*	*1.49*	*0.93*

SURVEY / SCORES

Modeling Languages, Model Representation, Maintenance and Evolution

	Importance		BSc Coverage		MSc Coverage		Importance minus:	
	Avg	Std.dev	Avg	Std.dev	Avg	Std.dev	BSc cov	MSc cov
4. Modeling Languages								
4.1. Language definition	3.68	*0.60*	2.27	*0.91*	2.94	*1.10*	1.42	0.75
* Metamodels	3.63	*0.73*	2.03	*1.02*	2.99	*1.14*	1.60	0.64
* Grammars	3.49	*0.74*	2.25	*1.03*	2.73	*1.16*	1.24	0.76
* Semantics (by e.g. Abstract State Machines or model transformations)	3.35	*0.81*	1.99	*0.84*	2.70	*1.11*	1.36	0.65
4.2. Types of modeling languages	3.67	*0.58*	2.40	*0.89*	2.89	*1.07*	1.28	0.78
* General purpose (GPL): UML+OCL, SysML	3.67	*0.63*	2.70	*0.89*	2.94	*1.03*	0.97	0.74
* Domain-specific (DSL): UML Profiles, Language Workbenches, ADLs, ...	3.57	*0.65*	1.76	*0.90*	2.80	*1.10*	1.81	0.77
4.3. Multiview Modeling	3.22	*0.79*	1.62	*0.77*	2.22	*1.14*	1.60	1.01
* Model viewpoints and views	3.24	*0.79*	1.65	*0.78*	2.20	*1.13*	1.59	1.04
* Correspondences among views	3.17	*0.85*	1.52	*0.76*	2.10	*1.10*	1.65	1.07
* Viewpoint consistency	3.14	*0.84*	1.47	*0.76*	2.01	*1.10*	1.67	1.13
* Viewpoint integration	3.11	*0.85*	1.41	*0.71*	1.97	*1.08*	1.70	1.15
5. Model Representation								
5.1. Concrete syntax	3.53	*0.73*	2.49	*1.06*	2.87	*1.05*	1.04	0.66
5.2 Physics of notations	2.80	*1.06*	1.56	*0.92*	1.97	*1.07*	1.24	0.84
5.3. Layout	2.99	*0.94*	1.63	*0.83*	2.10	*1.05*	1.36	0.89
5.4. Textual vs visual models	3.30	*0.81*	1.82	*0.86*	2.49	*1.11*	1.48	0.81
5.5. Animation	2.46	*0.96*	1.33	*0.67*	1.72	*1.01*	1.12	0.74
6. Model Maintenance and Evolution								
6.1. Model operations (diff, merge, refactoring)	3.40	*0.80*	1.35	*0.73*	1.95	*1.08*	2.04	1.45
6.2. Model Versioning	3.45	*0.76*	1.31	*0.64*	1.86	*1.04*	2.13	1.59
6.3. Model Migration	3.22	*0.85*	1.22	*0.50*	1.75	*0.97*	2.00	1.47

SURVEY / SCORES

Model Execution, Transformations

	Importance		BSc Coverage		MSc Coverage		Importance minus:	
	Avg	Std.dev	Avg	Std.dev	Avg	Std.dev	BSc cov	MSc cov
7. Model Execution								
7.1. Model simulation	3.30	*0.85*	1.49	*0.78*	1.95	*1.01*	1.81	1.36
* Model co-simulation (simulation of a hybrid model)	2.91	*0.95*	1.25	*0.62*	1.63	*0.95*	1.66	1.27
7.2. Execution strategies (sequential vs. parallel)	2.84	*0.95*	1.31	*0.64*	1.72	*1.00*	1.52	1.11
7.3. Model debugging and testing	3.37	*0.82*	1.41	*0.78*	1.87	*1.01*	1.97	1.51
8. Model Transformations								
8.1. Model transformation languages	3.46	*0.84*	1.52	*0.79*	2.56	*1.07*	1.93	0.90
* Syntax	3.33	*0.93*	1.50	*0.80*	2.59	*1.10*	1.83	0.74
* Semantics	3.43	*0.85*	1.39	*0.74*	2.48	*1.11*	2.03	0.94
8.2. Model transformation types	3.39	*0.86*	1.47	*0.74*	2.47	*1.09*	1.92	0.91
* Text-to-Model, M2M, M2T	3.47	*0.83*	1.53	*0.82*	2.60	*1.12*	1.94	0.87
* Exogenous vs. endogenous	3.24	*0.91*	1.29	*0.64*	2.28	*1.08*	1.95	0.95
* In-place vs. out-place	3.14	*0.97*	1.31	*0.72*	2.24	*1.11*	1.83	0.90
* Horizontal vs. vertical	3.12	*0.99*	1.29	*0.67*	2.11	*1.09*	1.82	1.00
* Uni-directional vs. bidirectional	3.20	*0.91*	1.25	*0.60*	2.06	*1.08*	1.95	1.14
* Syntactical vs. semantic	3.09	*0.99*	1.25	*0.60*	1.93	*1.04*	1.84	1.16
* Paradigm (declarative vs. operational)	3.22	*0.88*	1.30	*0.67*	2.14	*1.13*	1.92	1.08
8.3. Model transformation applications	3.49	*0.77*	1.45	*0.71*	2.41	*1.08*	2.04	1.08
* Model translation (synthesis, code generation, reverse engineering, migration, optimization, refactoring, refinement, adaptation) [10]	3.60	*0.73*	1.57	*0.80*	2.50	*1.10*	2.03	1.10
* Model merge	3.15	*0.89*	1.23	*0.55*	1.85	*0.99*	1.92	1.30
* Model differencing	3.15	*0.93*	1.26	*0.57*	1.82	*1.03*	1.89	1.33
* Model weaving	2.97	*0.99*	1.20	*0.52*	1.71	*0.96*	1.77	1.26
* Model synchronization	3.07	*0.92*	1.20	*0.52*	1.72	*0.87*	1.87	1.35
* Model interpretation (incl. execution)	3.24	*0.88*	1.30	*0.66*	2.02	*1.12*	1.94	1.22

SURVEY / SCORES

Further topics

	Importance		BSc Coverage		MSc Coverage		Importance minus:	
	Avg	Std.dev	Avg	Std.dev	Avg	Std.dev	BSc cov	MSc cov
9. Further topics								
9.1. How MBSE is used in application domains	3.58	*0.68*	1.58	*0.78*	2.10	*1.04*	2.00	1.48
* Automotive	3.28	*0.83*	1.42	*0.71*	1.87	*0.96*	1.86	1.41
* Cyber physical systems	3.31	*0.79*	1.43	*0.76*	1.93	*1.02*	1.87	1.37
* Industry 4.0	3.28	*0.83*	1.32	*0.67*	1.76	*0.94*	1.96	1.52
* Banking systems (e.g. modernization)	2.96	*0.94*	1.34	*0.70*	1.68	*0.91*	1.62	1.27
9.2. Advanced topics	2.69	*0.92*	1.22	*0.55*	1.57	*0.89*	1.47	1.12
* *Streaming model transformations*	2.42	*1.03*	1.14	*0.46*	1.38	*0.77*	1.28	1.05
* *Incremental transformations*	2.60	*0.99*	1.16	*0.52*	1.54	*0.87*	1.44	1.06
* *Uncertainty in modeling*	2.73	*1.01*	1.15	*0.47*	1.49	*0.89*	1.58	1.24
9.3. Application scenarios of MBSE	3.30	*0.78*	1.44	*0.80*	1.98	*1.02*	1.86	1.32
* Model based testing	3.28	*0.77*	1.46	*0.76*	2.03	*1.05*	1.81	1.24
* Model-based modernization	2.99	*0.87*	1.22	*0.59*	1.75	*0.95*	1.77	1.24
9.4. Engineering Best Practices	3.52	*0.73*	1.72	*0.93*	2.22	*1.12*	1.80	1.30
* How to use models to represent information systems	3.56	*0.71*	1.82	*0.99*	2.28	*1.16*	1.74	1.28
* How to use models to represent physical systems	3.40	*0.79*	1.52	*0.86*	1.98	*1.05*	1.88	1.42

MAIN FINDINGS

Topics with insufficient support

- Not important al all (the topic should not be included)
  - **More than 15%:** Animation (5.5), Streaming Transformations (9.2.1), Incremental Transformations (9.2.2)
  - **Between 10 and 15%:** Physics of notations (5.2), Further topics (9.2), and Uncertainty in Modeling (9.2.3)
- Low score in Importance (lack of support):
  - **Score below 2.5**: Animation (5.5), Streaming Transformations (9.2.1)
  - **Score below 2.75**: Further topics (9.2), Incremental Transformations (9.2.2), Uncertainty (9.2.3)
- No Opinion (lack of sufficient knowledge about the topic):
  - **Between 10% and 15%:** Streaming Transformations (9.2.1)
  - **More than 15%:** Physics of notations (5.2)

MAIN FINDINGS

Topics with low coverage

There is a clear correlation between importance and coverage

- important topics show higher coverage in existing curricula (the Pearson coefficient for the correlation between the assigned importance and the coverage at BSc level is 0.73 and 0.82 for the coverage at MSc level).
- significant exceptions are represented by model maintenance and execution: despite being considered fairly important, their coverage is rather low

Relative importance of subtopics

- The subtopics of
  - structural model analysis,
  - model transformation languages, and
  - MBE application domains

  are considered less relevant the the main topics.

Teaching metamodels and grammars

- there is a **discrepancy** in the community about when to teach meta-models and grammars, whether at BSc or MSc levels. This issue would require further analyses.

PROPOSAL

We excluded the topics that received little or no support

- Temporal model checking
- Physics of notation
- Animation

- Model transformation performance
- Streaming model transformations
- Incremental transformations

- Uncertainty in modeling

PROPOSAL

We excluded the topics that received little or no support

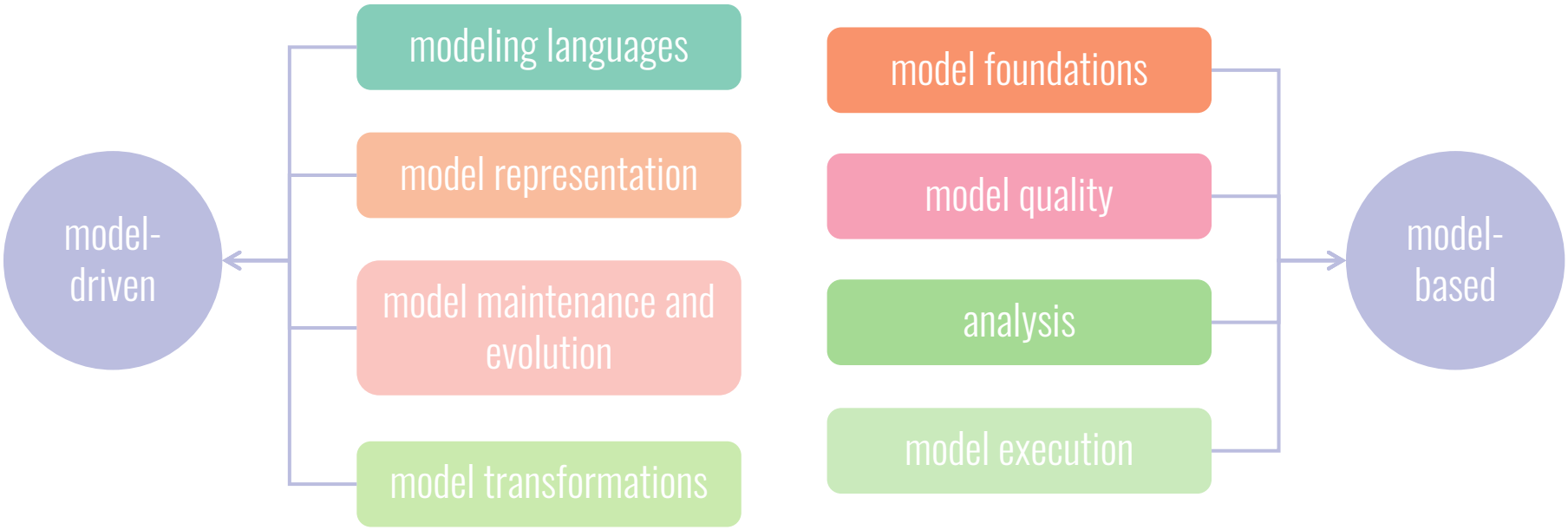
- Temporal model checking
- Physics of notation
- Animation

- Model transformation performance
- Streaming model transformations
- Incremental transformations

- Uncertainty in modeling

RELATIONS WITH SLEBOK AND SWEBOK

Topics might be clustered into **model-based** and **model-driven**



model-driven topics are related to language engineering

model-based topics are more related to software engineering

CONCLUSIONS

Aims

A list of topics for a MBEBOK proposal, it aims to

- characterize the contents and known practices, assist instructors and institutions that provide teaching courses on SE to develop their MBE curricula
- identify the core set of concepts that any MBE engineer should know, providing a common and consistent reference terminology

The intent is to help consolidate the field of MBE, clarify its scope with respect to other SE disciplines, and to improve the way it is currently taught

CONCLUSIONS

Future plans

Based on this proposal, we plan to start working on

the Guide to the MBE Body of Knowledge

which further develops these concepts and the practices that support the discipline.

It will supplement the SWEBOK and align with the current SLEBOK

- it will be made available in an open and collaborative platform
- educators and researchers of this community will be invited to contribute to this effort

Contact us at TheMBE- BOK@gmail.com

Thanks !