SM@RT

Smart Modeling for softw@re Research and Technology

CNRS IRIT Lab, Toulouse, France

https://www.irit.fr/smart

@SmartModelTeam

https://github.com/smart-researchteam

UNIVERSITÉ TOULOUSE Jean Jaurès

IRIT

The Team

SM@RT

Academic staff (3+1?):

- Jean-Michel Bruel, Prof., UT2J
- Pierre-Jean Charrel, Prof., UT2J
- Sophie Ebersold, Ass. Prof., UT2J
- ???, Ass. Prof., UT2J



Research staff (5):

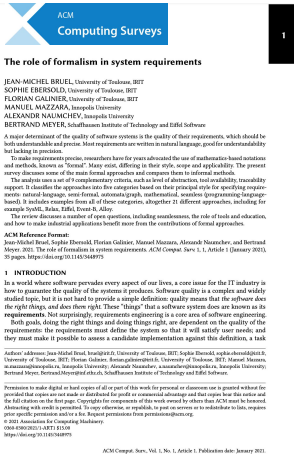
- Saloua Bennani, ATER
- Mohammad Chami, PhD Student
- Florian Galinier, PhD Student
- Hamna Aslam, PhD student
- Maria Naumcheva, PhD Student

Research Associates:

- Bertrand Meyer, Prof., SiT & Innopolis Univ.
- Benoit Combemale, Prof IRISA
- Thuy Ngyuen, EDF R&D
- Mahmoud El Hamlaoui, ENSIAS, Morocco

RE activities

- **Formal** Requirements
- Requirements **relationships**
- Formal requirements **Survey**
- **PEGS**: Formal OO Requirements approach

ACM Computing Surveys

1

The role of formalism in system requirements

JEAN-MICHEL BRUEL, University of Toulouse, IRIT
SOPHIE EBERSOLD, University of Toulouse, IRIT
FLORIAN GALINIER, University of Toulouse, IRIT
MANUEL MAZZARA, Innopolis University
ALEXANDR NAUMCHEV, Innopolis University
BERTRAND MEYER, Schaffhausen Institute of Technology and Eiffel Software

A major determinant of the quality of software systems is the quality of their requirements, which should be both understandable and precise. Most requirements are written in natural language, good for understandability but lacking in precision.

To make requirements precise, researchers have for years advocated the use of mathematics-based notations and methods, known as "formal". Many exist, differing in their style, scope and applicability. The present survey discusses some of the main formal approaches and compares them to informal methods.

The analysis uses a set of 9 complementary criteria, such as level of abstraction, tool availability, traceability support. It classifies the approaches into five categories based on their principal style for specifying requirements: natural-language, semi-formal, automata/graph, mathematical, seamless (programming-language-based). It includes examples from all of these categories, altogether 21 different approaches, including for example SysML, Reka, Eiffel, Event-B, Alloy.

The review discusses a number of open questions, including seamlessness, the role of tools and education, and how to make industrial applications benefit more from the contributions of formal approaches.

ACM Reference Format:
Jean-Michel Bruel, Sophie Ebersold, Florian Galinier, Manuel Mazzara, Alexandr Naumchev, and Bertrand Meyer. 2021. The role of formalism in system requirements. *ACM Comput. Surv.* 1, 1, Article 1 (January 2021), 35 pages. https://doi.org/10.1145/3448975

1 INTRODUCTION

In a world where software pervades every aspect of our lives, a core issue for the IT industry is how to guarantee the quality of the systems it produces. Software quality is a complex and widely studied topic, but it is not hard to provide a simple definition: quality means that the *software does the right things, and does them right*. These "things" that a software system does are known as its **requirements**. Not surprisingly, requirements engineering is a core area of software engineering.

Both goals, doing the right things and doing things right, are dependent on the quality of the requirements: the requirements must define the system so that it will satisfy user needs; and they must make it possible to assess a candidate implementation against this definition, a task

Authors' addresses: Jean-Michel Bruel, bruel@irit.fr, University of Toulouse, IRIT; Sophie Ebersold, sophie.ebersold@irit.fr, University of Toulouse, IRIT; Florian Galinier, florian.galinier@irit.fr, University of Toulouse, IRIT; Manuel Mazzara, m.mazzara@innopolis.ru, Innopolis University; Alexandr Naumchev, a.naumchev@innopolis.ru, Innopolis University; Bertrand Meyer, Bertrand.Meyer@inf.ethz.ch, Schaffhausen Institute of Technology and Eiffel Software.

0360-0300/2021/1-ART1 $15.00
https://doi.org/10.1145/3448975

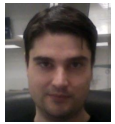
Handbook of Requirements and Business Analysis

Bertrand Meyer
Shaffhausen Institute of Technology

Version: 2021-05-20
Draft, do not circulate
Change log is on next page

Formal requirements

- **Specification Drivers**
  - Eiffel Requirements Patterns



- **RSML**: Requirements Specific Modeling Language
  - DSL for requirements (https://gitlab.com/Ynigvi/RSML)

Requirements **relationships**

SM@RT

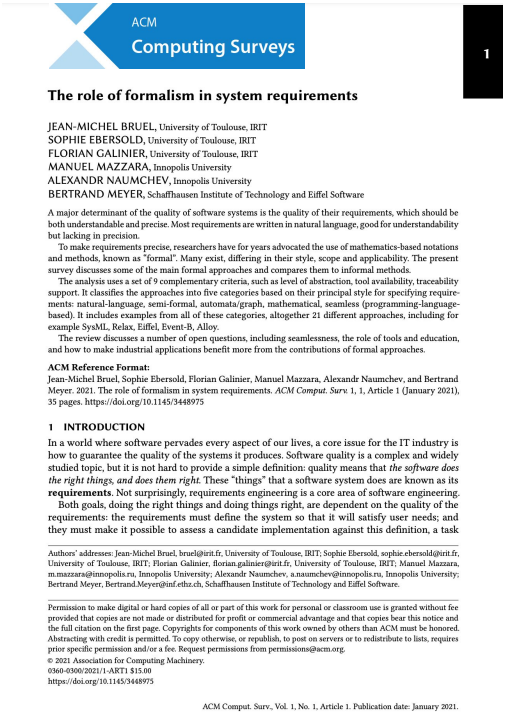
- Disjoins (X || Y)
- Belongs (X ⊆ Y)
- Repeats (X ⇔ Y)
- Contradicts (X ⊕ Y)
- Extends (X > Y)
- Excepts (X \\ Y)
- Constraints (X Y)
- Characterizes (X → Y)

Formal requirements **survey**

4 years

8 reviewers

85p. answers



ACM
Computing Surveys

The role of formalism in system requirements

JEAN-MICHEL BRUEL, University of Toulouse, IRIT
SOPHIE EBERSOLD, University of Toulouse, IRIT
FLORIAN GALINIER, University of Toulouse, IRIT
MANUEL MAZZARA, Innopolis University
ALEXANDR NAUMCHEV, Innopolis University
BERTRAND MEYER, Schaffhausen Institute of Technology and Eiffel Software

A major determinant of the quality of software systems is the quality of their requirements, which should be both understandable and precise. Most requirements are written in natural language, good for understandability but lacking in precision.

To make requirements precise, researchers have for years advocated the use of mathematics-based notations and methods, known as "formal". Many exist, differing in their style, scope and applicability. The present survey discusses some of the main formal approaches and compares them to informal methods.

The analysis uses a set of 9 complementary criteria, such as level of abstraction, tool availability, traceability support. It classifies the approaches into five categories based on their principal style for specifying requirements: natural-language, semi-formal, automata/graph, mathematical, seamless (programming-language-based). It includes examples from all of these categories, altogether 21 different approaches, including for example SysML, Relax, Eiffel, Event-B, Alloy.

The review discusses a number of open questions, including seamlessness, the role of tools and education, and how to make industrial applications benefit more from the contributions of formal approaches.

ACM Reference Format:
Jean-Michel Bruel, Sophie Ebersold, Florian Galinier, Manuel Mazzara, Alexandr Naumchev, and Bertrand Meyer. 2021. The role of formalism in system requirements. *ACM Comput. Surv.* 1, 1, Article 1 (January 2021), 35 pages. https://doi.org/10.1145/3448975

1 INTRODUCTION

In a world where software pervades every aspect of our lives, a core issue for the IT industry is how to guarantee the quality of the systems it produces. Software quality is a complex and widely studied topic, but it is not hard to provide a simple definition: quality means that *the software does the right things, and does them right.* These "things" that a software system does are known as its **requirements**. Not surprisingly, requirements engineering is a core area of software engineering.

Both goals, doing the right things and doing things right, are dependent on the quality of the requirements: the requirements must define the system so that it will satisfy user needs; and they must make it possible to assess a candidate implementation against this definition, a task

Authors' addresses: Jean-Michel Bruel, bruel@irit.fr, University of Toulouse, IRIT; Sophie Ebersold, sophie.ebersold@irit.fr, University of Toulouse, IRIT; Florian Galinier, florian.galinier@irit.fr, University of Toulouse, IRIT; Manuel Mazzara, m.mazzara@innopolis.ru, Innopolis University; Alexandr Naumchev, a.naumchev@innopolis.ru, Innopolis University; Bertrand Meyer, Bertrand.Meyer@inf.ethz.ch, Schaffhausen Institute of Technology and Eiffel Software.

0360-0300/2021/1-ART1 $15.00
https://doi.org/10.1145/3448975

PEGS: Formal OO Requirements approach

- Bertrand Meyer upcoming book!

Start-up!

- Tech. Transfer
- RSML Industrialisation

http://spilen.fr/

RE activities

- **Formal** Requirements => 2 PhDs + 1 Start-up
- Requirements **relationships** => A taxonomy of relationships
- Formal requirements **Survey** => Published in **ACM Surveys**
- **PEGS**: Formal OO Requirements approach => A **book** in 2021

SM@RT

Open thoughts

- $$ => hard to get **projects**/fundings
- Small community => **Collaboration**
- **Open data** => shared repo of requirements?