A metamodel to guide a requirements elicitation process for embedded systems

Tarcísio Pereira, Aêda Sousa, Reinaldo Silva, Deivson Albuquerque, Fernanda Alencar, and Jaelson Castro

tarcisio.couto@ifsertao-pe.edu.br
tcp@cin.ufpe.br

Centro de Informática UFPE
Universidade Federal de Pernambuco
INSTITUTO FEDERAL
Sertão Pernambucano
Campus Petrolina Zona Rural

QUATIC 2018

Outline

- Motivation;
- Goals;
- Metamodel Development Process;
- Requirements Elicitation Process;
- Conclusions and Future Work.

Motivation

- Broy (1999) defines Embedded System as a *system that regulates a physical device by sending control signals to actuators in reaction to input signals provided by its users and by sensors capturing the relevant state parameters of the system.*

- Embedded Systems (ES) are present in different domains such as automation technology, automotive, avionics, energy technology or medical technology.

Motivation

- In everyday life, people are **dependent** on several services supported by software, many of them **transparent** to the final user.

- The majority of ES are **less visible**, and they run in **engines, brakes, seat belts, airbag, and audio system** in your car.

- Furthermore, such systems are designed to repeatedly carry out a specific function, keeping its operation under different constraints from the ones of general purpose systems [2], [4]

Motivation

- According to Broy (1999) [1], in the embedded system domain, more than 50% of the problems occur when the system is delivered. [misconceptions in capturing requirements];

- These shortcomings are the result of **inappropriate Requirements Engineering (RE) tasks**, resulting **in incomplete requirements, incorrect elicitation and specification, high complexity, and economic or human loss.**

- A requirements engineering process is crucial to meet time, cost, and quality goals [7];

Motivation

- Important point;
- Some studies **[8-10]** have investigated the concepts that should be considered during Embedded Systems development. However, these studies did not captured the core concepts and appropriate evidence.

- Additionally, a systematic investigation was not performed;

Motivation

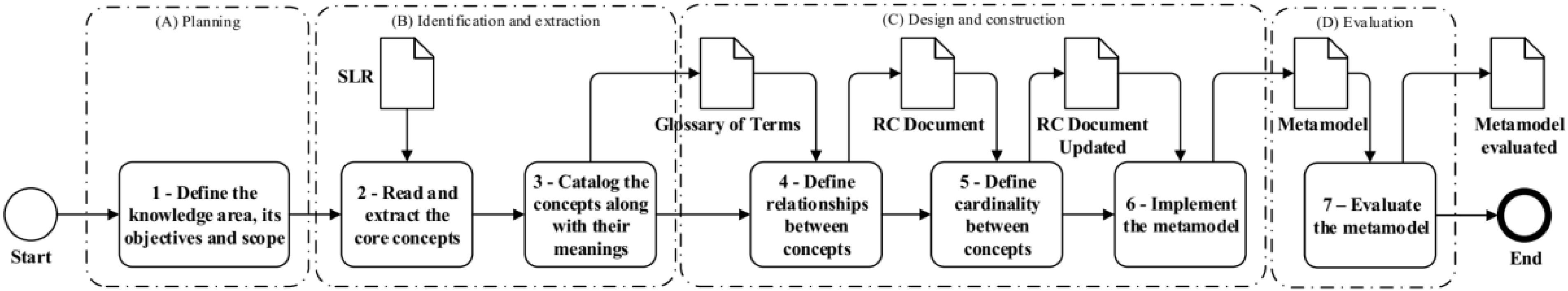
- In fact, there is much **confusion** among requirements engineers and stakeholders due to the different kinds of information that need to be managed.

- A metamodel that capture embedded systems concepts would constitute a significant step forward to improve the requirements quality of embedded systems.

Goals

- Update and evaluate a metamodel (MM4ES) for embedded systems;
- Develop a Requirements Engineering Process for Embedded Systems (REPES) based on the metamodel concepts; In this research, we focused on elicitation actions.

Metamodel Development Process



Adapted from Usman et al. [19]

(A) Planning

- Knowledge area
- What is the core set of requirements engineering information that should be specified by requirements engineers in the development of embedded systems?;

(B) Identification and extraction

- Data source: articles from our SLR;
- Inspection to identify the embedded systems concepts;
- Concepts definition;

(C) Design and Construction

- Define relationships and cardinality between concepts;
- Implement the metamodel;

(D) Evaluation

- First evaluation – The metamodel was analyzed three times by a **domain expert** to check if the concepts were correct from his point of view in the context of medical device development;
- Second evaluation - we tried to demonstrate the ***utility of the metamodel by tracing the results of the elicitation process actions to the embedded systems concepts*** using the metamodel as a basis.

(C) Design and (

- Define relationshi
- Implement the me

Class name	Relation	Cardinality	Class name
Embedded System	hasDomainKnowledge	1..1	Domain Knowledge
Embedded System	hasAbstractionLevel	1..0*	Abstraction Level
Embedded System	hasSoftware	1..1*	Software
Embedded System	hasStakeholder	1..1*	Stakeholder
Embedded System	hasContext	0..1	Context
Embedded System	hasHardware	1..0*	Hardware
Embedded System	hasAction	1..0*	Action
Embedded System	hasBusiness	1..1	Business
Embedded System	hasEnvironmentRequirements	1..0*	Environment Requirements
Software Requirements	defines	1*..1*	Software
Functional Requirements	is-a		Software Requirement
Non-Functional Requirements	is-a		Software Requirement
Hardware	hasHardwareRequirements	1..1*	Hardware Requirements
Hardware Requirements	defines	1..1*	Hardware Device
Sensor	is-a		Hardware Device
Sensor	monitors	1..0*	Control Variables
Actuator	is-a		Hardware Device
Actuator	generates	1*..1*	Action
Environment	hasEmbeddedSystem	1..0*	Embedded System

(D) Evaluation

- First evaluation – The metamodel was analyzed three times by a **domain expert** to check if the concepts were correct from his point of view in the context of medical device development;
- Second evaluation - we tried to demonstrate the ***utility of the metamodel by tracing the results of the elicitation process actions to the embedded systems concepts*** using the metamodel as a basis.

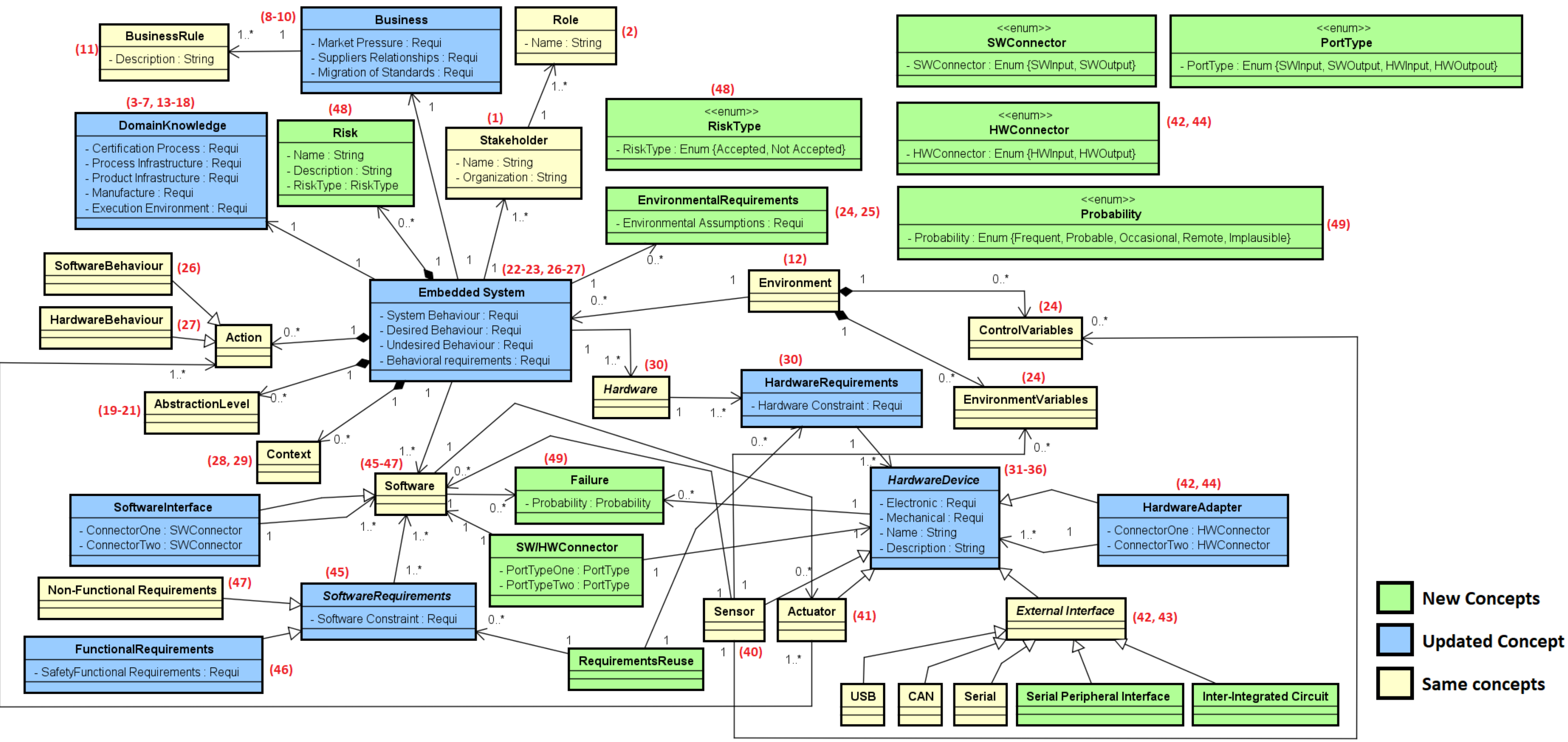
(C) Design and Construction

- Define relationships and cardinality between concepts;
- Implement the metamodel;

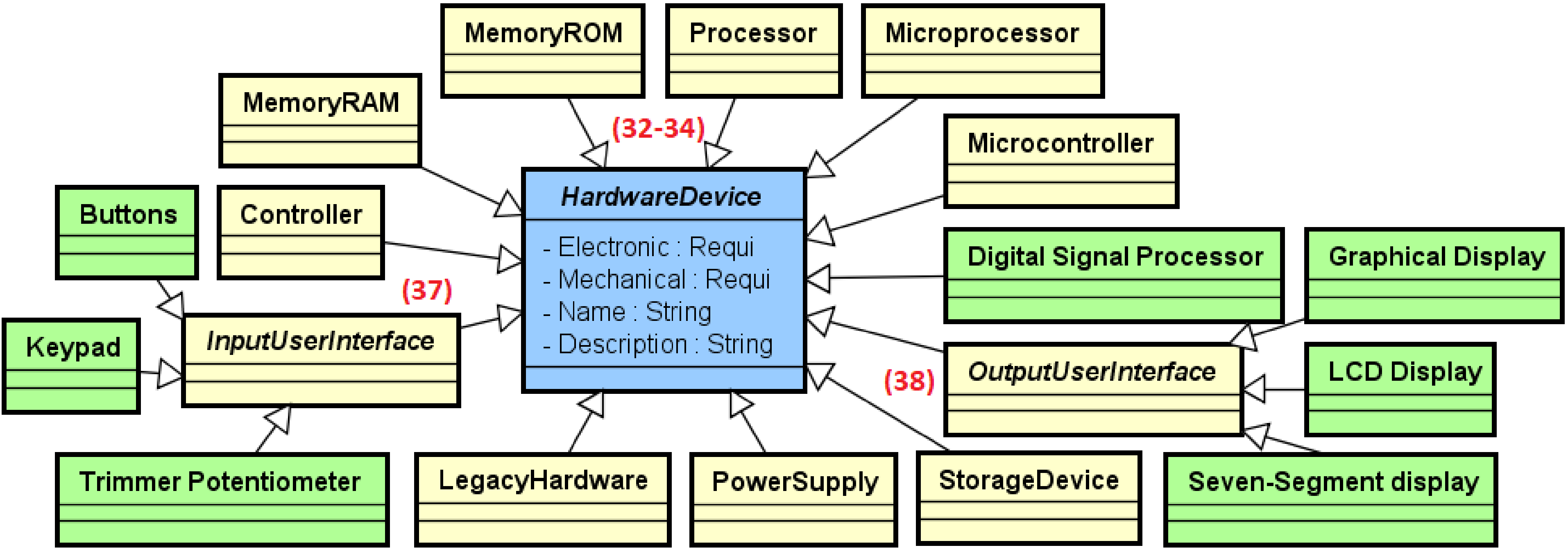
(D) Evaluation

- First evaluation – The metamodel was analyzed three times by a **domain expert** to check if the concepts were correct from his point of view in the context of medical device development;
- Second evaluation - we tried to demonstrate the ***utility of the metamodel by tracing the results of the elicitation process actions to the embedded systems concepts*** using the metamodel as a basis.

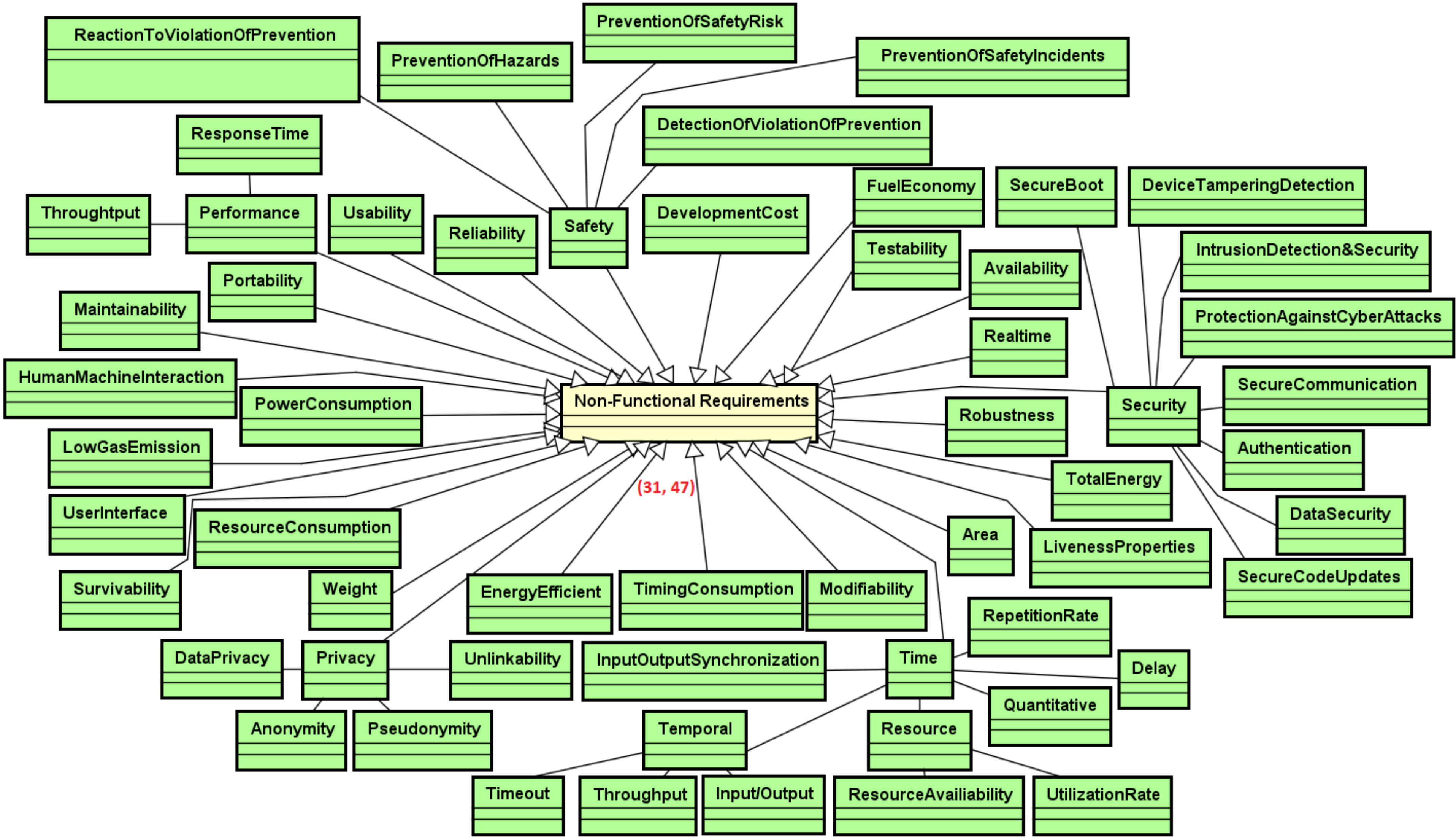
Metamodel for Embedded Systems – MM4ES



Metamodel for Embedded Systems – MM4ES



Metamodel for Embedded Systems – MM4ES



Process Development Methodology

- (1) Knowledge acquisition
- (2) Problem definition
- (3) Mapping of concepts
- (4) Identification of information sources
- (5) Definition of process design
- (6) Development of the process

Process Development Methodology

- (1) Knowledge acquisition → SLR
- (2) Problem definition
- (3) Mapping of concepts
- (4) Identification of information sources
- (5) Definition of process design
- (6) Development of the process

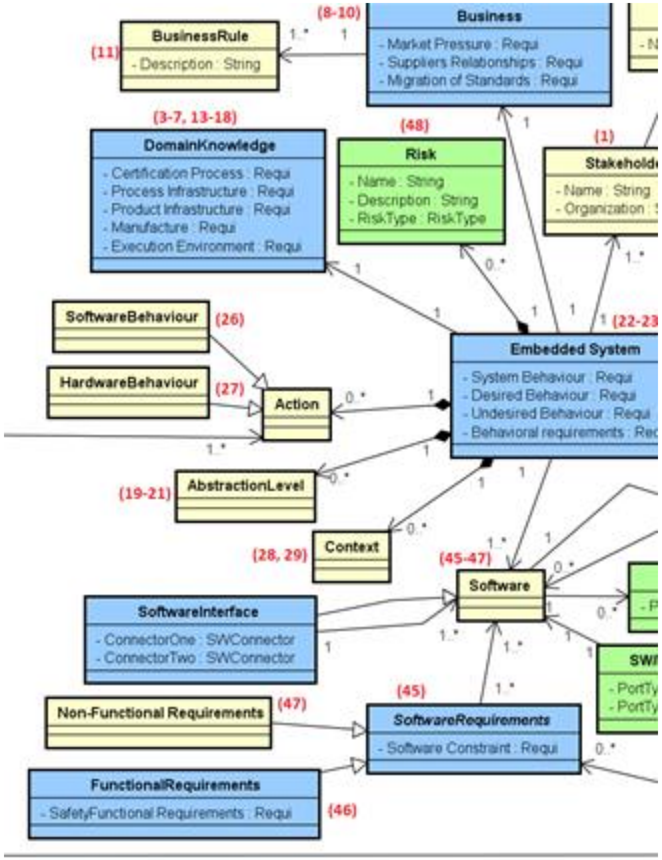
Process Development Methodology

(i) the lack of a requirements engineering process. (ii) what should be considered to develop the RE process?

- (1) Knowledge acquisition
- (2) Problem definition
- (3) Mapping of concepts
- (4) Identification of information sources
- (5) Definition of process design
- (6) Development of the process

Process Development Methodology

- (1) Knowledge acquisition
- (2) Problem definition
- (3) Mapping of concepts
- (4) Identification of information sources
- (5) Definition of process design
- (6) Development of the process



Process Development Methodology

Sub-process	Definition of behavioral requirements
Metamodel Concepts	Action,Software Behavior, HardwareBehavior
State of the Art (SLR)	S01,S20,S24,S27,S28,S29,S31,S44,S53,S56,S60,S62,S67,S73,S75,S79
Requirements STD	ISO/IEC 12207, ISO/IEC 15288, ISO/IEC 15289, ISO/IEC 29148, Uni-REPM

- (1) Knowledge acquisition
- (2) Problem definition
- (3) Mapping of concepts
- (4) Identification of information sources
- (5) Definition of process design
- (6) Development of the process

Metamodel concepts

- Studies of the SLR
- RE Standards

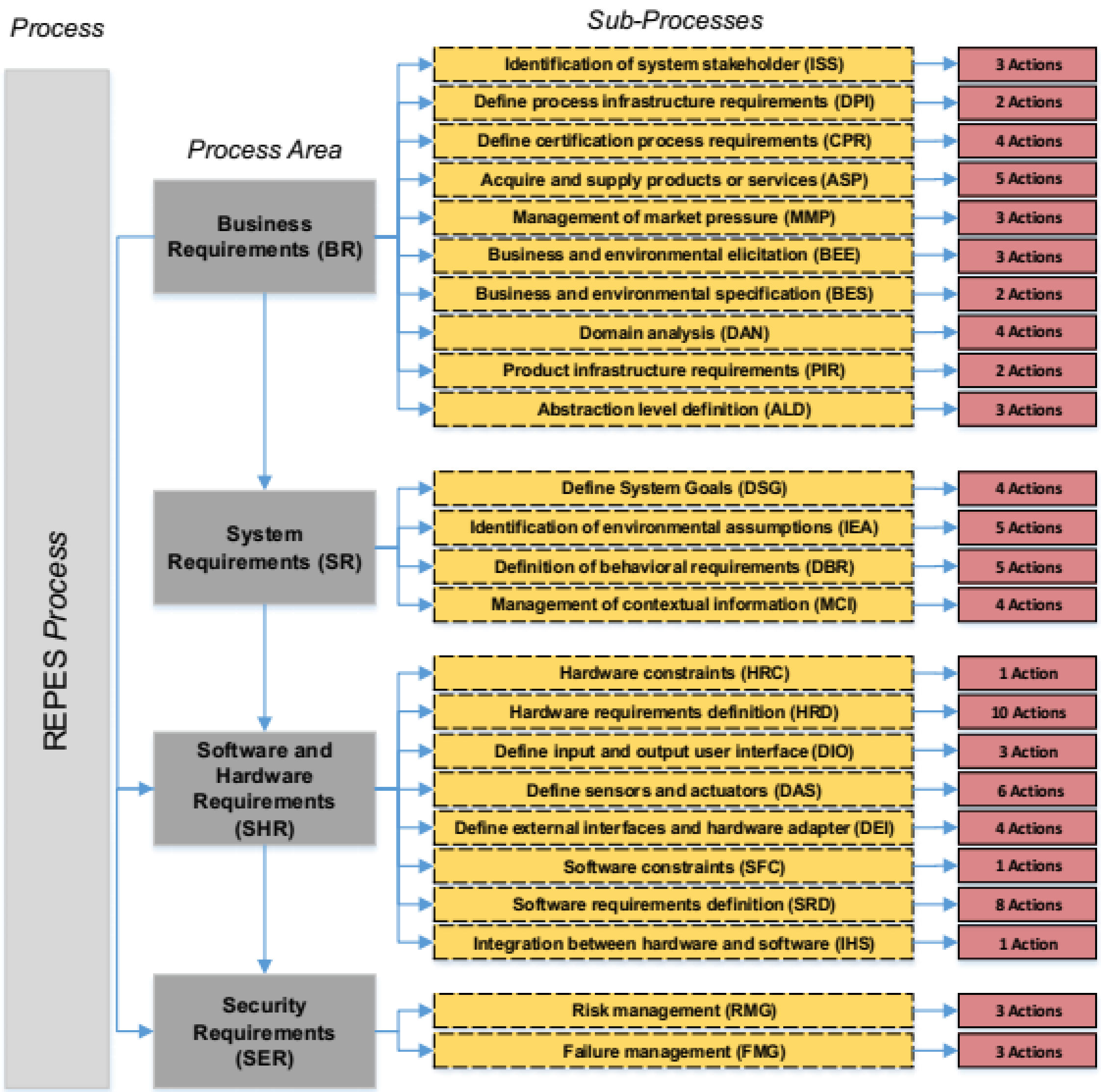
Process Development Methodology

- (1) Knowledge acquisition
- (2) Problem definition
- (3) Mapping of concepts
- (4) Identification of information sources
- (5) Definition of process design
- (6) Development of the process

The REPES Process

- Using the process, ES engineers can manage the elicitation activity in an organized way;

- 24 sub-processes;
- 89 actions;
- 49 actions related to requirements elicitation.



The REPES Process – Elicitation Actions

#	ID	Description
	SR	System Requirements
	SR.DSG	Define System Goals
22	SR.DSG.a1	Provide short statements describing what the system must accomplish
23	SR.DSG.a3	Elicit a set of system goals from stakeholder's needs
	SR.IEA	Identification of Environmental Assumptions
24	SR.IEA.a1	Establish and document the monitored and controlled variables
25	SR.IEA.a5	Establish a list of devices that monitor the monitored and controlled variables
	SR.DBR	Definition of Behavioral Requirements
26	SR.DBR.a1	Provide a set of software behaviors to document the actions the software should perform
27	SR.DBR.a2	Provide a set of hardware behaviors to document the actions the hardware should perform
	SR.MCI	Management of Contextual Information
28	SR.MCI.a1	Analyze the input documents to identify the contexts that can affect the system operation
29	SR.MCI.a2	Identify and document statements, facts, and variables
	SHR	Software and Hardware Requirements
	SHR.HRC	Hardware Constraints
30	SHR.HRC.a1	The constraints for each hardware device previously identified are defined and documented
	SHR.HRD	Hardware Requirements Definition
31	SHR.HRD.a2	Elicit and specify a set of non-functional requirements that the hardware devices must fulfill
32	SHR.HRD.a3	Provide an overview of the hardware components to be used in the development of an embedded system
33	SHR.HRD.a4	Elicit the mechanical requirements
34	SHR.HRD.a5	Elicit electrical requirements
35	SHR.HRD.a6	Identify and document the microcontroller of the embedded system based on the results of previous actions
36	SHR.HRD.a9	Elicit a set of manufacture requirements
	SHR.DIO	Define Input and Output User Interface
	SHR.DIO.a1	Define and document interface standards
37	SHR.DIO.a2	Define and document a set of input user interface
38	SHR.DIO.a3	Define and document a set of output user interface
	SHR.DSA	Define Sensors and Actuators
39	SHR.DSA.a1	Define and document a set of sensors and actuators standards
40	SHR.DSA.a2	Identify and document a set of sensors
41	SHR.DSA.a5	Identify and document a set of actuators
	SHR.DEI	Define External Interfaces and Hardware Adapter
42	SHR.DEI.a1	Identify and document a set of external interfaces and hardware adapters standards
43	SHR.DEI.a2	Identify and document a set of external interfaces
44	SHR.DEI.a3	Identify and document the hardware adapters
	SHR.SFC	Software Constraints
45	SHR.SFC.a1	Identify and document the software constraints
	SHR.SRD	Software Requirements Definition
46	SHR.SRD.a1	Elicit and document a set of functional software requirements
47	SHR.SRD.a5	Elicit and document a set of non-functional requirements
	SER	Security Requirements
	SER.RMG	Risk Management
48	SER.RMG.a1	Identify and document potential risks
	SER.FMG	Failure Management
49	SER.FMG.a1	Identify, classify, and document potential software and hardware failures

Process Usage

- Use of the process: when an organization decides to introduce requirements engineering activities in its embedded systems development process;

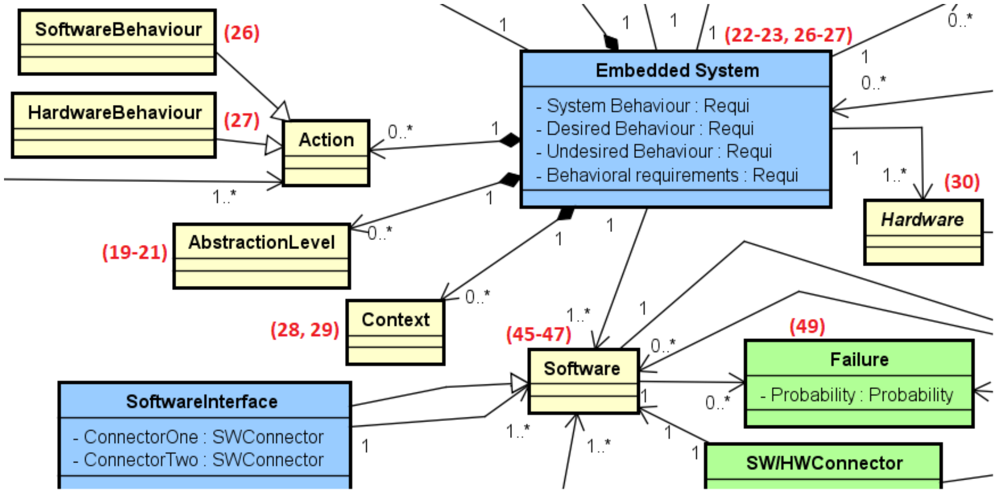
- The actions can be performed in a sequential way to get a document with the requirements elicited;

- We used a real world scenario related to a medical device as a proof of concept. The goal is to demonstrate the utility of the process by presenting examples as results of the actions;

Process Usage

Sub-process	Definition of behavioral requirements
Metamodel Concepts	Action,Software Behavior, HardwareBehavior
State of the Art (SLR)	S01,S20,S24,S27,S28,S29,S31,S44,S53,S56,S60,S62,S67,S73,S75,S79
Requirements STD	ISO/IEC 12207, ISO/IEC 15288, ISO/IEC 15289, ISO/IEC 29148, Uni-REPM

- We used a set of documents as input for the requirements elicitation process;
- Scenario: Patient-controlled analgesic (PCA) infusion pump;



26 - SR.DBR.a1 - (i) after the start button has been pushed, a timer counter shall be displayed, and (ii) when the infusion is in progress, a boolean signal shall be displayed.

27 - SR.DBR.a2 - (i) after the button A has been pushed, a red light shall be lit, and (ii) after the dose button has been pushed, two beeps shall be sounded, and the pump will begin delivering the demand dose.

	SR.DBR	Definition of Behavioral Requirements
26	SR.DBR.a1	Provide a set of software behaviors to document the actions the software should perform
27	SR.DBR.a2	Provide a set of hardware behaviors to document the actions the hardware should perform

Conclusion

- We propose a metamodel for embedded systems to describe the main concepts that should be elicited and specified by requirements engineers; and
- We used the metamodel to guide the development of a requirements elicitation process;
- The process describes actions for the identification and definition of embedded systems requirements.

Conclusion

- The main contributions of this research are the following:
  1. a knowledge-based metamodel;
  2. a reference metamodel that can be updated in future works;
  3. a metamodel that can be used as input for model transformation;
  4. an elicitation process;
  5. the process can be used as a guide to assess the requirements activities of organizations.

Limitations

1. Only a metamodel cannot address all characteristics of an ES for a specific domain;
2. The metamodel and the requirements elicitation process was not evaluated in the industry yet; and
3. A threat may be the selection of actions that are included in the REPES process since they were based on the metamodel concepts and RE standards.

Research directions

1. How can we evaluate the completeness of the proposed metamodel?
2. How can we extend the metamodel to represent the specific characteristics of the different domains of an ES?
3. How can we evaluate whether the process has sufficient coverage of actions?
4. How can we validate the usefulness and ease of use of the process?
5. How can we develop a CASE tool to support the process?
6. How can we conduct a comparative analysis with other studies using the same case study (PCA)?

References

[1] M. Broy and T. Stauner, “Requirements engineering for embedded systems,” Informationstechnik und Technische Informatik, vol. 41, pp. 7–11, 1999.

[7] E. Sikora, B. Tenbergen, and K. Pohl, “Industry needs and research directions in requirements engineering for embedded systems,” Requirements Engineering, vol. 17, no. 1, pp. 57–78, 2012.

[8] J. Ossada, “Gerse: Requirements elicitation guide for small and medium size companies, in portuguese: Gerse: Guia de elicitação de requisitos para pequenas e medias empresas,” Piracicaba, Sao Paulo, Brazil, 2010.

[9] G. Kainz, C. Buckl, S. Sommer, and A. Knoll, “Model-to-metamodel transformation for the development of component-based systems,” in 13th International Conference on Model Driven Engineering Languages and Systems. Springer, 2010, pp. 391–405.

[10] H. Fennel, S. Bunzel, H. Heinecke, J. Bielefeld, S. Furst, K.-P. Schnelle, ¨ W. Grote, N. Maldener, T. Weber, F. Wohlgemuth et al., “Achievements and exploitation of the autosar development partnership,” Convergence, 2006.

References

[13] T. Arpinen, T. Ham ¨ al ¨ ainen, and M. H ¨ annik ¨ ainen, “Meta-model and ¨ uml profile for requirements management of software and embedded systems,” EURASIP Journal on Embedded Systems, vol. 2011, no. 1, p. 592168, 2011.

[14] M. Li, F. Batmaz, L. Guan, A. Grigg, M. Ingham, and P. Bull, “Modelbased systems engineering with requirements variability for embedded real-time systems,” in Fifth Model-Driven Requirements Engineering Workshop at 23rd IEEE International Requirements Engineering Conference, 2015.

[15] H. Dubois, M.-A. Peraldi-Frati, and F. Lakhal, “A model for requirements traceability in a heterogeneous model-based design process: Application to automotive embedded systems,” in 15th ieee international conference on engineering of complex computer systems, 2010.T.

[19] M. Usman, R. Britto, J. Borstler, and E. Mendes, “Taxonomies in software engineering: A systematic mapping study and a revised taxonomy development method,” Information and Software Technology, 2017.

Petrolina - Brazil

Thank you!

Caruaru - Brazil

A metamodel to guide a requirements elicitation process for embedded systems

Tarcísio Pereira, Aêda Sousa, Reinaldo Silva, Deivson Albuquerque, Fernanda Alencar, and Jaelson Castro

tarcisio.couto@ifsertao-pe.edu.br
tcp@cin.ufpe.br

Centro de Informática U·F·P·E
Universidade Federal de Pernambuco
INSTITUTO FEDERAL
Sertão Pernambucano
Campus Petrolina Zona Rural