



**HAL**  
open science

## Requirements Elicitation for a Light Control System

Maritta Heisel, Jeanine Souquières

► **To cite this version:**

Maritta Heisel, Jeanine Souquières. Requirements Elicitation for a Light Control System. Seminar No. 99241, chloss Dagstuhl, Requirements Capture, Documentation & Validation, E. Börger (Università di Pisa, I), B. Hörger (Daimler-Benz, Ulm, D), D. Parnas (McMaster University, CAN), D. Rombach (Universität Kaiserslautern, D), 1999, Dagstuhl, Wadern, Germany, 3 p. inria-00098762

**HAL Id: inria-00098762**

**<https://inria.hal.science/inria-00098762>**

Submitted on 26 Sep 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Requirements Elicitation for a Light Control System

Maritta Heisel  
Fakultät für Informatik  
Universität Magdeburg  
D-39016 Magdeburg, Germany  
Fax: (49)-391-67-12810  
heisel@cs.uni-magdeburg.de

Jeanine Souquière  
LORIA—Université Nancy2  
B.P. 239 Bâtiment LORIA  
F-54506 Vandœuvre-les-Nancy, France  
Fax: (33)-3-83-41-30-79  
souquier@loria.fr

April 1, 1999

## Abstract

We use a systematic approach to clarify and analyze the requirements for the light control system.

## 1 Objectives

The objectives of our method for requirements elicitation are:

- understand the problem
- fix the vocabulary (only one name per concept)
- disambiguate the requirements
- find incoherences
- find missing requirements
- establish an adequate starting point for a formal specification of the software system

## 2 Method

We have developed a method to perform the early phases of the software lifecycle in a systematic way, carefully distinguishing requirements elicitation and specification development. The starting point of the requirements elicitation phase is an informal requirements document. The method begins with a brainstorming process where the application domain and the requirements are described in natural language. This informal description is then transformed into a formal representation. On the formal representation, consistency analyses are performed. Their purpose is to obtain a consistent set of requirements. The methods are expressed as *agendas* [Hei98].

The method for requirements elicitation is described in [HS98b], the method to detect interactions in requirements is described in [HS98a]. We give a brief overview of the steps:

### Method for Requirements Elicitation

1. Introduce the domain vocabulary.
2. State the facts, assumptions, and requirements concerning the system.

3. List all relevant events that can happen in connection with the system, and classify them (Who is in control of the event? Who can observe the event?).
4. List the system operations that can be invoked by the users.
5. Formalize the facts, assumptions, and requirements as constraints on the possible traces of system events.

The first four steps are usually performed in parallel.

### Method to Incorporate Single Constraints

When a new requirement is added to a set of already existing requirements, it should be analyzed if interactions (i.e., inconsistencies or unwanted behavior) can occur. Hence, Step 5 of the method for requirements elicitation is performed as follows:

1. Formalize the new constraint as a predicate on system event traces.
2. Give a schematic expression of the constraint.
3. Update the tables of semantic relations. These are:
  - (a) Necessary conditions for events: Which predicates on the system state must be true for the event to happen?
  - (b) Events establishing predicate literals: Which predicates are established after an event has happened?
  - (c) Implication relations between predicates
4. Determine interaction candidates. (There exists an automatic procedure for this purpose.)
5. Decide if there are interactions of the new constraint with the determined candidates.
6. If an interaction occurs, take one of the following actions:
  - correct a fact
  - relax a requirement
  - strengthen an assumption

## 3 Documents Produced

The result of our work consists of several documents:

1. an entity-relationship diagram to fix the domain vocabulary and to express *static* aspects of the system
2. a list of events, together with their classification, concerning the *reactive* part of the system
3. a list of system operations, concerning the *transformational* part of the system
4. a list of functions and predicates on the system state that were introduced when formalizing the requirements
5. tables of semantic relations between event and predicate symbols
6. a formalization of the requirements, expressed as constraints on admissible system event traces
7. schematic expressions for the requirements, together with interaction candidates

## 4 Results

The goal of our investigation was not only to perform the case study but also to validate our method.

### 4.1 Results Concerning the Method

The method turned out to be well suited to achieve our goals (see Section 1). Even though the steps were not performed separately and in exactly the given order, they structure the requirements elicitation process and help focus attention to crucial points. In particular,

- several repetitions of the various steps were necessary
- static, reactive and transformational parts of the system are well distinguished
- traceability of requirements is guaranteed

### 4.2 Results Concerning the Informal Requirements Document

Using our systematic approach, we found the following flaws in the informal requirements document:

- incoherent vocabulary
- glossary partially inadequate
- usage of too low-level concepts
- confusion of different aspects (hardware and system behavior)
- confusion between functional and non-functional requirements
- ambiguous requirements
- unsatisfiable requirements
- incoherent requirements
- missing requirements

After exchanging several emails with the organizers, we were able to

- identify a coherent subset of the given vocabulary
- resolve some of the incoherences between requirements
- point out open problems that should be solved before a realization of the system should be undertaken.

## References

- [Hei98] Maritta Heisel. Agendas – a concept to guide software development activities. In R. N. Horspool, editor, *Proc. Systems Implementation 2000*, pages 19–32. Chapman & Hall London, 1998.
- [HS98a] Maritta Heisel and Jeanine Souquière. A heuristic approach to detect feature interactions in requirements. In K. Kimbler and W. Bouma, editors, *Proc. 5th Feature Interaction Workshop*, pages 165–171. IOS Press Amsterdam, 1998.
- [HS98b] Maritta Heisel and Jeanine Souquière. Methodological support for requirements elicitation and formal specification. In *Proc. 9th International Workshop on Software Specification and Design*, pages 153–155. IEEE Computer Society Press, 1998.