PUC

ISSN 0103-9741

Monografias em Ciência da Computação
nº 34/92

A Case Study on Requirements Recovery from Structured Specifications

Julio C. S. P. Leite

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900
RIO DE JANEIRO - BRASIL

PUC RIO - DEPARTAMENTO DE INFORMÁTICA ISSN 0103-9741

Monografias em Ciência da Computação, Nº 34/92
Editor: Carlos J. P. Lucena Dezembro, 1992

A Case Study on Requirements Recovery from Structured Specifications*

Julio C. S. P. Leite

* This work has been sponsored by the Secretaria de Ciência e Tecnologia da Presidência da República Federativa do Brasil.

In charge of publications:
Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC Rio — Departamento de Informática
Rua Marquês de São Vicente, 225 — Gávea
22453-900 — Rio de Janeiro, RJ
Brasil
Tel. +55-21-529 9386 Telex +55-21-31048 Fax +55-21-511 5645
E-mail: rosane@inf.puc-rio.br
techrep@inf.puc-rio.br (for publications only)

A Case Study on Requirements Recovery from Structured Specifications

Julio Cesar Sampaio do Prado Leite
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
R. Marquês de S. Vicente 225 Rio de Janeiro 22453
Brasil

May 1992

Abstract

Requirements recovery is a process to produce software requirements from specifications. In order to test a proposed method for requirements recovery we conducted an experience involving six systems analysts and ten structured analysis specifications. The data already available shows several aspects of both the specifications and the recovered requirements. We include the raw data in this report.

1 Introduction

We have been investigating a reverse engineering strategy for discovering requirements. Departing from structured analysis specifications [Gane 79] we recast requirements as a set of requirement sentences with a lexicon [Leite 92]. We worked with the hypothesis that a restricted natural language description is better than a specification, if we want to validate the requirements with users.

Our scenario is the growing need and trend to use more formal languages [Spivey 90] during software specifications. As software must be maintained, it is appealing to recast the existing semi-formal specifications in more formal languages. Although this can be done departing from the semi-formal specifications themselves, we believe that it is better to start formalization from a set of requirements easily readable by users.

The representation that is been used for the User Requirements Document is a combination of simple sentences with a language extended lexicon (LEL) [Leite 89]. The idea of requirement as a simple sentence is well defined by Burstin [Burstin 84], that considers it the basic unit of information in requirements. A LEL is a hyperdocument that uses a special hypertext system, HyperLex©[1], to implement its structure and to give support to its use. The LEL proposal is strongly influenced by other areas of study [Eco 79] [Carnap 56].

[1] ©JCSPL and APMF

2 Requirements Statements and the LEL

There are three classes of requirements statements: input, output and state change. Each requirement must follow the following two structures:

> *The system must* + [verb + object | verb phrase] + [agent complement | null] + [condition | null].
>
> *The system must* + [verb + object | verb phrase] + [agent complement | null] + { a) condition-1, b) condition-2, ... condition-n}

LEL is a set oriented description . It has three different entities:

- signs,
- notions, and
- behavioral responses.

These entities are represented as a hypertext as the two principles below are followed.

- When describing a notion or a behavioral response **maximize** the use of signs (**Sign**) of the language extended lexicon. We call this the *principle of circularity.*
- When describing a notion or a behavioral response **minimize** the use of signs exterior (¬**Sign**) to the target application. Using external signs make sure they belong to the basic vocabulary of the natural language in use, also, as much as possible, have a clear mathematical representation (eg. Set, belongs, intersection, function). We call this the *principle of minimal vocabulary.*

The use of LEL with the statements is very simple.

> Each basic component in the statement is highlighted, and will be an entry in the LEL.

The basic components are: verb, object, agent complement, and condition(s). As such each entry must follow the rules of LEL formation.

As an example of this representation we use the line division problem presented by Meyer [Meyer 85] in his article criticizing natural language descriptions. The natural language description of the problem was first presented by Naur and later improved by Gerhart. Using our representation the list of requirements is as in Figure 1 and in Figures 2,3,4, and 5[2] we present part of the LEL for Naur's problem.

[2]The LEL entries are presented in HyperLex, and as such notions and behavioral response are in Portuguese. Notion is *noção* and Behavioral Response is *impacto*. *Retorna* is return and *Termina* is end.

1. The system must **read** an **input text** from a **file**. (INPUT CLASS)
2. The system must **reject** an **input text** in which there are **words** greater than **MAXPOS**. (INPUT CLASS)
3. The system must **write** an **output text** into a **file**. (OUTPUT CLASS)
4. The system must **split** the **input text** into **lines** such that:
   (a) a **split happens** at a **blank** or at a **NL** character,
   (b) each **splited line** is **filled as far as possible**,
   (c) each **splited line length** is less than **MAXPOS** characters,
   (d) the order of **words** in **input text** is the same as in the **output text**. (STATE CHANGE CLASS)

Figure 1: The Requirements List

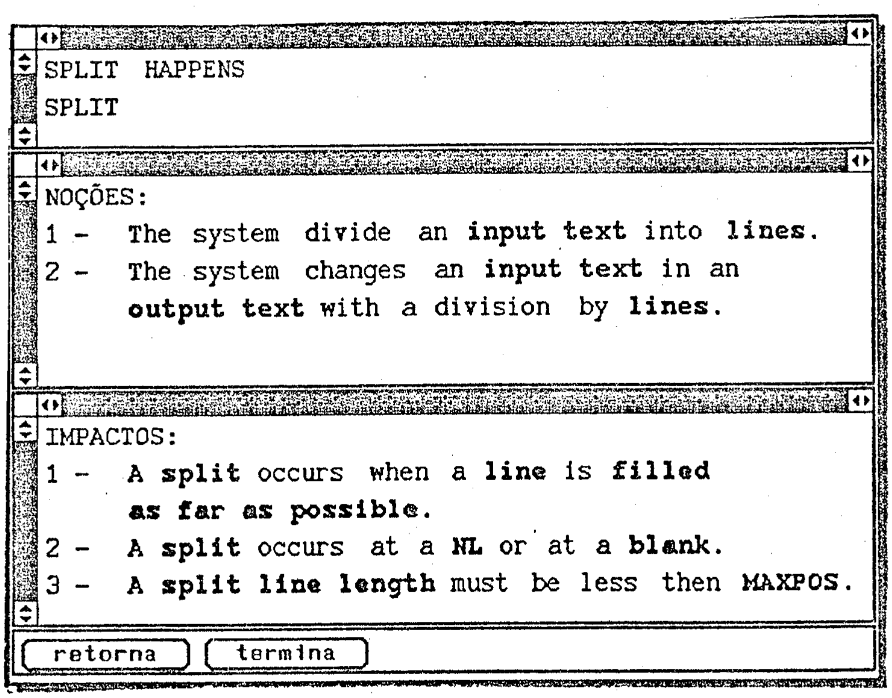

Figure 2: Split, a Sign in the LEL

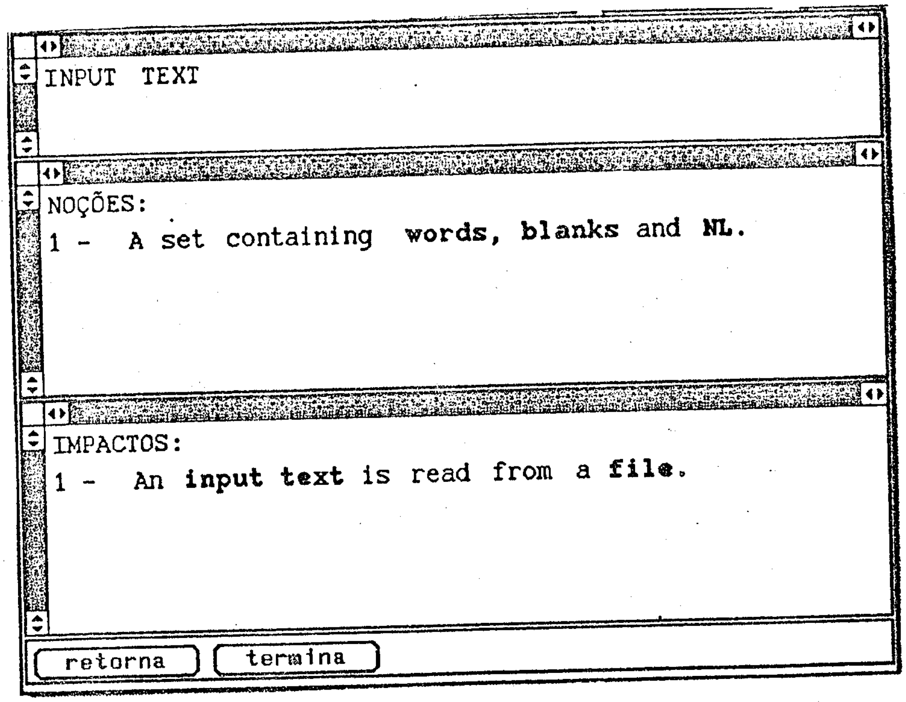

Figure 3: Input Text, a Sign in the LEL

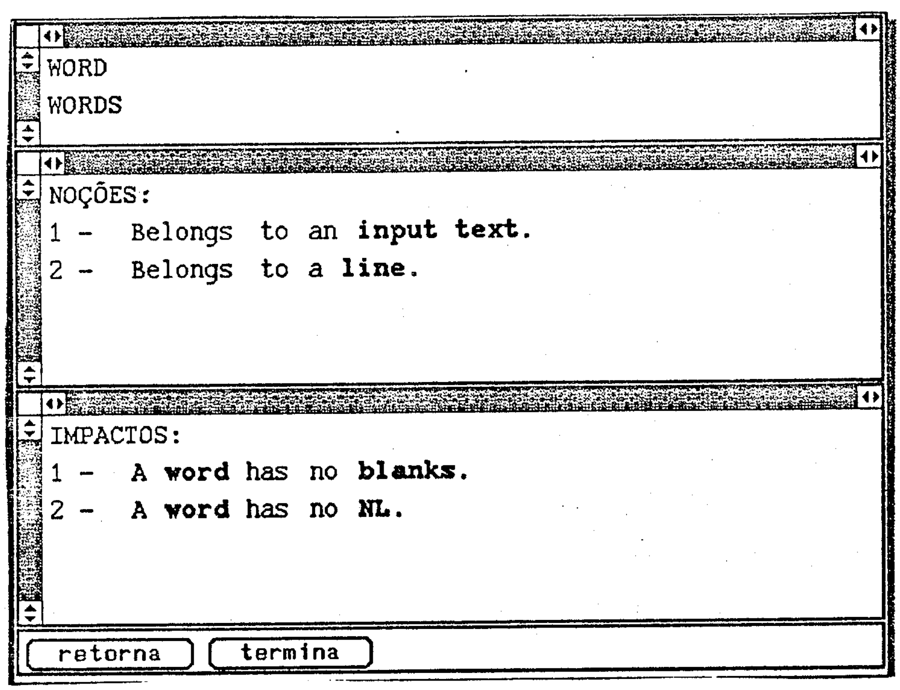

Figure 4: Word, a Sign in the LEL

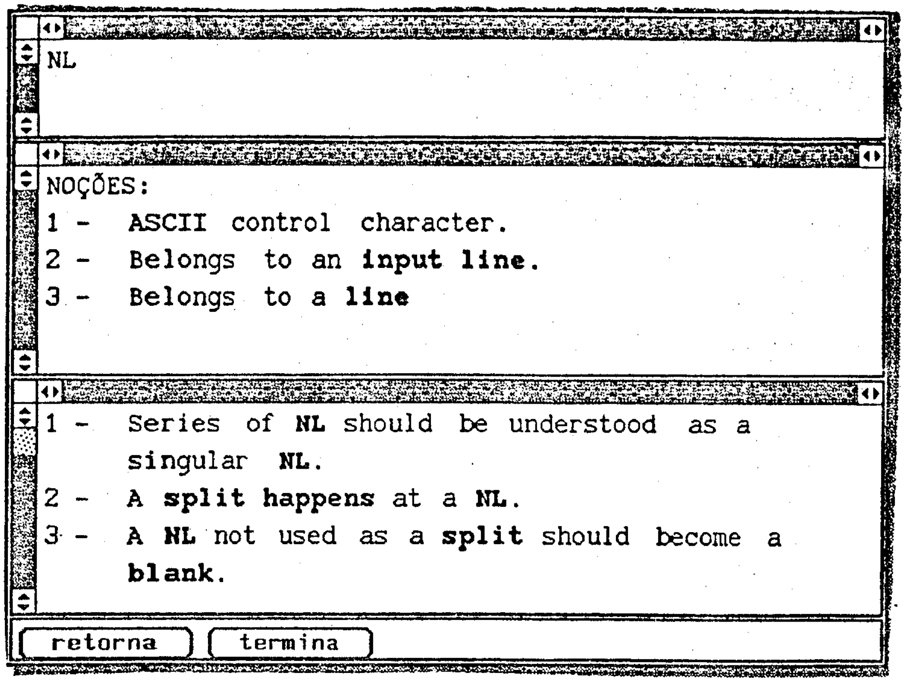

Figure 5: NL, a Sign in the LEL

3 Recovering Requirements from Structured Specifications

The procedure we are testing is very straightforward. Given a DFD use the efferent activities as the main element for an output class requirement. Use the afferent activities for input class and the central activities for the state change class. The agent complement is given by external entities, and conditions are usually previous activities. Note that activities are the *bubbles*, and that not all efferent, afferent or central bubbles will yield a requirement. The data dictionary as well as the informal description are used in the task of producing the LEL. This task also uses the DFD diagrams to get hints for the behavioral responses.

4 The Case Study, and Its Data

Our case study used as basis a set of 10 structured analysis specifications produced by data processing professionals as a final project in an extension course on software engineering. Five of those specifications were related to Video Store administrative systems and five related to Gas Station administrative systems. None of them departed from the same situation, they were done independently. Each specification had its own informal description. Each specification is composed of an informal description, a set of DFD diagrams, a Data Dictionary and a set of process specifications. In average the specifications had one and a half page of informal description, 7 DFD diagrams (including context), and 50 entries in DFD[3]. The limits were: maximum (3 pages of informal description, 9 diagrams and 111 entries), minimum (1 page, 3

[3] We did not list process specifications (logic) as they were not used in the method

diagrams, and 7 entries).

The objective of the experiment was twofold, help the validation of the recovering strategy and compare the same system requirements recovered by different actors. The design of the experiment was as follows. Each specification was recovered by two subjects, using the heuristics suggested above, and by limiting the task to seven hours for each specification. The author controled the experiment.

The details of the experiment are been used to improve the recovering procedure. We measured that although there is a high, in the average, ratio of similarity between recovered requirements, it was below we first thought. The ratio was around 62%[4], that is 62% of the requirements and the entries in LEL were almost identical. This data may reveal two things. First our recovered procedure is still too flexible. Second, different subjects, even restricted by some imposing rules, have different perceptions [Leite 91].

We also observed that all actors reinforced our hypothesis of readability of the recovered requirements if compared to the specification itself. Another observation was about the quality of the specifications. In building the list and the lexicon, some actors had more troubles than others. There were specifications that could not provide the information needed to perform the recovery, mostly the information to fill the LEL.

Below we list the raw data acquired in this experience. The data will be organized by system. GS stands for gas station, and VS for video store. ID is the size of informal description, DFD is the size of the set of diagrams, DD is the number of entries in the Data Dictionary, RS is the number of requirements recovered and LEL is the number of entries in the lexicon.

- VS-1: ID – 1, DFD – 3, DD – 22.
  - Subject A: RS – 15, LEL – 20.
  - Subject B: RS – 12, LEL – 13.
- Hit Ration: RS-A/RS-B – 12, LEL-A/LEL-B – 10.
- VS-2: ID – 1, DFD – 6, DD – 85.
  - Subject A: RS – 10, LEL – 20.
  - Subject B: RS – 24, LEL – 31.
- Hit Ration: RS-A/RS-B – 7, LEL-A/LEL-B – 6.
- VS-3: ID – 2, DFD – 7, DD – 66.
  - Subject A: RS – 17, LEL – 27.
  - Subject B: RS – 16, LEL – 24.
- Hit Ration: RS-A/RS-B – 9, LEL-A/LEL-B – 12.
- VS-4: ID – 1, DFD – 4, DD – 7.
  - Subject A: RS – 19, LEL – 20.
  - Subject B: RS – 17, LEL – 25.

[4]This ratio does not count one pair that had presented an extreme variance

- Hit Ration: RS-A/RS-B – 9, LEL-A/LEL-B – 15.
- VS-5: ID – 2, DFD – 5, DD – 15.
  - Subject A: RS – 9, LEL – 17.
  - Subject B: RS – 14, LEL – 24.
- Hit Ration: RS-A/RS-B – 8, LEL-A/LEL-B – 14.
- GS-1: ID – 2, DFD – 4, DD – 44.
  - Subject A: RS – 18, LEL – 29.
  - Subject B: RS – 14, LEL – 27.
- Hit Ration: RS-A/RS-B – 14, LEL-A/LEL-B – 17.
- GS-2: ID – 2, DFD – 3, DD – 36.
  - Subject A: RS – 9, LEL – 15.
  - Subject B: RS – 9, LEL – 15.
- Hit Ration: RS-A/RS-B – 3, LEL-A/LEL-B – 2.
- GS-3: ID – 1, DFD – 6, DD – 53.
  - Subject A: RS – 10, LEL – 12.
  - Subject B: RS – 15, LEL – 28.
- Hit Ration: RS-A/RS-B – 6, LEL-A/LEL-B – 9.
- GS-4: ID – 1, DFD – 3, DD – 79.
  - Subject A: RS – 8, LEL – 19.
  - Subject B: RS – 19, LEL – 25.
- Hit Ration: RS-A/RS-B – 8, LEL-A/LEL-B – 13.
- GS-5: ID – 2, DFD – 8, DD – 111.
  - Subject A: RS – 11, LEL – 27.
  - Subject B: RS – 29, LEL – 35.
- Hit Ration: RS-A/RS-B – 9, LEL-A/LEL-B – 19.

5 Conclusion and Further Study

It is our intention to pursue on designing software engineering experiments so as to gain more insights on the process and the product level knowledge involved in software production. Our results here show that specifications' quality can be measured by trying to recover their requirements. The experiments also pointed out that different subjects do present different perceptions of a specification, even when following certain guidelines. It is also our intention to investigate the application of viewpoint resolution [Leite 91] in the results of our case study.

6 Acknowledgement

Ana Paula Gilvaz, Ricardo Castro, Carlos Duarte, Adolfo Mathias, José Geraldo Araújo and Paulo Cerqueira produced the samples of our experience. Maria Margarida Vaz helped in the analysis of the samples.

References

[Burstin 84] Burstin, M. *Requirements Analysis of Large Software Systems.* Ph.D. dissertation, Tel - Aviv University, Israel, Jul. 1984.

[Carnap 56] Carnap, R. The Methodological Character of Theoretical Concepts, *In Feigl, H. Scriven, M. Minnesota Studiesin the Philosophy of Science.* Minneapolis, 1956.

[Eco 79] Eco, U. *A Theory of Semiotics*, University Press, Bloomington, 1979.

[Gane 79] Gane, C., Sarson, T., *Structured Systems Analysis: Tools and Techniques*, Prentice-Hall Inc., 1979.

[Leite 89] Leite, J.C.S.P. Elicitation of Application Languages, *In Monografias em Ciência da Computação*, PUC-RIO, no. 30, 1989.

[Leite 90] Leite, J.C.S.P. and Franco, A.P.M. O Uso de Hipertexto na Elicitação de Linguagens da Aplicação. *IV Simpósio Brasileiro de Engenharia de Software*, São Paulo, (Out 1990).

[Leite 91] Leite, J.C.S.P. and Freeman, A. P. Requirements Validation Through Viewpoint Resolution, *In IEEE Trans. on Software Engineering*, Vol. 17, N. 12, Dec. , 1991.

[Leite 92] Leite, J.C.S.P. Enhancing The Semantics of Requirements Statements, *In Monografias em Ciência da Computação*, PUC-RIO, no. 22, 1992.

[Meyer 85] Meyer, B. On Formalism in Specifications, *IEEE Software*, Jan. 1985.

[Spivey 90] Spivey, J. M., Specifying a Real-Time Kernel, *In IEEE Software*, Vol. 7, N. 5, Sep. 1990. Communicating Ideas. In *Tutorial on Design Techniques*, Freeman and Wasserman, Eds., IEEE Computer Society Press, Long Beach, CA 1980, pp. 107-125.