The World's a Stage: A Survey on Requirements Engineering Using a Real-Life Case Study

Karin Koogan Breitman[1] **Julio Cesar Leite[1]** **Anthony Finkelstein[2]**

[1]Departamento de Informática
Pontifícia Universidade Católica
R. Marquês de São Vicente, 225
Gávea – Rio de Janeiro – RJ 22453-900
+55 21 529-9462
[karin,julio]@inf.puc-rio.br

[2]Computer Science Department
University College London
Gower Street
London WC1E 6BT
acwf@cs.city.ac.uk

PUC-RioInf.MCC07/98 March, 1998

Keywords: Requirements Engineering, Viewpoints, Real-Life Case Study, Software Engineering

Abstract:
In this report we present an analysis of a case study drawn from a real life setting, namely the London Ambulance Service System (LAS). We depart from a very detailed description of a real system that had problems and point out to specify research results that could have been applied in order to minimize some of them. This case study provides a unique opportunity to look at the results from research on the Requirements Engineering field through the prism of a real-life case study. In this light, a contribution of this work is showing the applicability of research findings in a real life setting. A natural outcome of our effort is a set of arguments sustaining the cause for Requirements Engineering research.

This report is mostly based on the case study information provided by the Report of the Inquiry of the LAS, that provides a detailed account of the development and implementation of a real system and its surroundings. We survey some of the most recent methods and techniques and, at the same time, point out to real situations where they could have been used and, perhaps minimized some of the perceived problems. The main contribution of this paper is pointing out situations in which results from recent research efforts in Requirements Engineering could, to some degree, be put to practice. As researchers, one of our major concerns is recognizing problems in real world practice and identifying candidates for further research.

Palavras-chave: Engenharia de Requisitos, ViewPoints, Estudo de Caso de Situação Real, Engenharia de Software

Resumo:
Neste relatório apresentamos uma análise de estudo de caso extraído de uma situação real,o Sistema de Serviços de Ambulâncias de London (LAS). A partir de uma descrição detalhada de um sistema real com problemas, apontamos resultados de pesquisa que poderiam ter sido aplicados para minimizar alguns desses problemas. Esse estudo de caso oferece uma oportunidade única de examinar resultados de pesquisa no campo de Engenharia de Requisitos sob a luz de uma situação real. Uma das contribuições deste trabalho é mostrar a aplicabilidade de resultados de pesquisa em uma situação real. Um resultado natural desse esforço é um conjunto de argumentações a favor da causa da pesquisa em Engenharia de Requisitos.

Este relatório é baseado principalmente nas informações do estudo de caso fornecidas pelo "Report of the Inquiry" do LAS, que oferece uma descrição detalhada do desenvolvimento e implementação de um sistema real e seu contexto. Pesquisamos alguns dos métodos e técnicas mais recentes e, ao mesmo tempo, apontamos situações reais onde eles poderiam ter sido utilizados e ter minimizado alguns dos problemas percebidos. A principal contribuição deste artigo é apontar situações em que os resultados de esforços recentes em Engenharia de Requisitos poderiam ser postos em prática. Como pesquisadores, nossa principal preocupação é reconhecer problemas na prática do mundo real e identificar candidatos para pesquisas futuras.

The World's a Stage: A Survey on Requirements Engineering Using a Real-Life Case Study

Karin Koogan Breitman[1] **Julio Cesar Leite[1]** **Anthony Finkelstein[2]**

[1]Departamento de Informática
Pontifícia Universidade Católica
R. Marquês de São Vicente, 225
Gávea – Rio de Janeiro – RJ 22453-900
+55 21 529-9462
[karin,julio]@inf.puc-rio.br

[2]Computer Science Department
University College London
Gower Street
London WC1E 6BT
acwf@cs.city.ac.uk

In this report we present an analysis of a case study drawn from a real life setting, namely the London Ambulance Service System (LAS). We depart from a very detailed description of a real system that had problems and point out to specify research results that could have been applied in order to minimize some of them. This case study provides a unique opportunity to look at the results from research on the Requirements Engineering field through the prism of a real-life case study. In this light, a contribution of this work is showing the applicability of research findings in a real life setting. A natural outcome of our effort is a set of arguments sustaining the cause for Requirements Engineering research.

This report is mostly based on the case study information provided by the Report of the Inquiry of the LAS, that provides a detailed account of the development and implementation of a real system and its surroundings. We survey some of the most recent methods and techniques and, at the same time, point out to real situations where they could have been used and, perhaps minimized some of the perceived problems. The main contribution of this paper is pointing out situations in which results from recent research efforts in Requirements Engineering could, to some degree, be put to practice. As researchers, one of our major concerns is recognizing problems in real world practice and identifying candidates for further research.

1. Introduction

1.1 Objectives

In this report we present an analysis of a case study drawn from a real life setting, namely the London Ambulance Service System (LAS). This case study has been the subject of a number of publications since the 1993 Report on the Inquiry into the London Ambulance [RI-LAS93] was made public [Luff94], [Jirotka95] [Finkelstein96] . Our work differs from previous attemps by providing links between problems observed from the report to research topics in the field of Requirements Engineering. We depart from a very detailed description of a real system that had problems and point out to specific research results that might have been applied in order to minimize some of the them. In this light, a contribution of this work is showing the applicability of research findings in a real life setting. A natural outcome of our effort is a set of arguments sustaining the cause for Requirements Engineering research.

The events surrounding the development and implementation of the LAS System had a great repercussion in the community. The system was recently used as the case study at the 8th International Workshop on Software Specification and Design. This workshop has established a tradition of using case studies to conduct discussions over selected themes. The discussion was anchored in the Report of the Inquiry on the London Ambulance Service (LAS) and both this document and the proceedings of the workshop were made available to software community at large [Kramer96]. We therefore take for granted that members of the requirements engineering commutiny are familiar with the LAS system to some degree.

The idea for this paper sprung from the analysis of the Report of the Inquiry of the LAS, that provides a detailed account of the development and implementation of a real system and its surroundings [RI-LAS93]. This case study provides a unique opportunity to look at the results from research on the requirements engineering field through the prism of a real-life case study. We attempt to survey some of the most recent methods and techniques and, at the same time, point out real situations where they could have been used and perhaps minimized some of the perceived problems. We are, by no means, trying to find mistakes or solutions to the problems the LAS system has gone through. On the contrary, we are using their former experience as a means to acquire a better understanding of how events take place in real life, thus situating scenarios where research results could be used to benefit. Another goal of this study is to exemplify the ocurrence of problems narrated by the liteture in a real life setting.

In the case of the LAS as of many others, [Brooks75] we will observe that no single cause can be pointed out as the sole responsible for the system breakdown, but the sum of a series of minor problems could be held responsible. We strongly believe that with the advances of software research in general and requirements engineering in particular, many of such problems could have been minimized or even eliminated. It is in this spirit that we conducted this study.

In a sense the Report of the Inquiry is similar to the famous Knuth report on errors on T_eX [Knuth89] looks at the problems of a complex software written by him and basically stand alone, whereas the report looks at problems of a complex system written by many in a situation where several technologies would integrate. While Knuth's report is very focused and investigates software alone, the LAS report is very broad and also considers external aspects of software development. However, both deal with software errors and are based on well defined error recording procedures.

We belive that this report should be of interest to the software community at large, and to the requirements engineering community in particular. A survey on recent findings on the requirements engineering field is conducted aiming to bring to context new methods, techniques and tools that should be of help to both reaserchers and practitioners. The main contribution of this paper, though, is in pointing out situations in which results from recent research efforts in requirements engineering could, to some degree, be put to practice. As researchers, one of our major concerns is recognizing problems in real world practice and identifying candidates for further research.

1.2 Sources of Information

This study was basically performed based on the information provided by the Report of the Inquiry of the LAS[RI-LAS93]. To the best of our knowledge, the report is the lengthier and more detailed account of the events surrounding the development of the LAS system. The inquiry report uses a very much managerial oriented view. The problems reported are framed mostly from the point of view of an experienced project manager. Nonetheless the inquiry goes beyond simple project managing. A broad analysis was taken an there are flashes of the whole context. The inquiry report team was composed of a chief executive, a chief conciliation officer, a senior computer audit partner and the support of a least another executive (former chief executive).

The report is very well organized, divided in six chapters and each paragraph is numbered to ease further referencing. The report is also available electronically[1].The initiative of publicizing and making it available on the Internet comes from Anthony Finkelstein and John Dowell that presented the LAS. We make many references to the contents of the report throughout this study, we quote the report whenever we see fit. We have imported various fragments of the Report so that it is not necessary to resource to the Report of the Inquiry of the LAS to understand this paper. Quotations from the Report will appear as blocks and in italics. The numbers after the blocks are the paragraph numbers around which the report is organized.

Other references provide insights based on the report itself. Those include [Luff94] and [Jirotka95] that analyze social aspects related to the LAS and the documentation around the 8th IWSSD, [Finkelstein96], [Kramer96] and [IWSSD96].

1.3 Organization of this paper

Our work method consisted of the careful reading of the report on the inquiry into the LAS [RI-LAS93] and the available literature on the subject [Luff94], [Jirotka95] [Finkelstein96]. Using the report on the inquiry as the main source, we proceeded by taking down passages that, we believed, illustrated the more significant events that determined the downfall of the system. We have came up to 56 different such passages. We then classified them into different groups, our criteria being based on the report of the inquiry itself, previous analysis on the subject conducted by other authors and our knowledge on recent requirements engineering literature. It is important to notice that we had no preconceptions on the classification, prior to the selection of the passages. The organizing criteria emerged from scrutiny of the selected passages, in a “on-the-fly” fashion.

The resulting groups were mapped into four different areas, namely multitude of opinions, evolution, environmental aspects and non-functional aspects. Those are detailed as follows. It is important to notice that some passages were classified in more than one area, demonstrating an intense correlation among the areas.

- Multitude of Opinions – the social organization surrounding the development of the LAS system was a very complex one. A great number of actors were involved, to some degree, with the system and its deployment. To name just a few there were ambulance crews, ground staff and management, each party with different opinions in relation to the LAS system. This section takes a deeper look into this situation and its consequences. There are 19 passages classified in this area.

---

[1] [1] ftp://ftp.cs.city.ac.uk/pub/requirements/lascase0.9.ps.gz

- Evolution – in this section we take a closer look at the system in order to uderstand the evolution of events that ultimately resulted in its downfall. Based on some key issues we tried to uncover the rationale behind design decisions and take a look at documentation. Issues such as changes to the software, specifications (both hardware and software) and technology trade-offs are among the ones addressed . There are 17 passages classified in this area.

- Environmental aspects – this section concentrates in the social aspects surrounding the development and deployment of the system. Issues such as company policies, regulations and the impact the system might have in the organization are central to this discussion. There are 19 passages classified in this area.

- Non-Functional aspects – this section is concerned with aspects that. Although not directly related to the system, played an important role in its outcome. We believe one of the most significant ones is the communication problem. The system did not take into account that in an environment as crowded as London , radio communications are sometimes difficult, and that resulted in a series of reported malfunctions. Less critical issues, such as cost and the trade-off between desired performance and user interface, among other problems, are identified and addressed in this section. There are 28 passages classified in this area.

2. Multitude of Opinions

A careful reading of the Report on the Inquiry on the London Ambulance Service [RI-LAS93] will indicate that the system dealt with was a very complex one composed of many and, perhaps, equally complex parts. Those include managerial, organizational and computational aspects and deploy a variety of different resources (people, software, hardware, vehicles, etc..). References were found in the report that indicated that each one of these sub-systems is intimately related to one another and, at the same time, possess unique characteristics that distinguish them from the rest. Our primary interest here is investigate where and if viewpoint oriented techniques could have been used to help elucidating some of the problems described by the report. We must first understand that the software system is only a part of a much larger system, the computer aided dispatch (CAD). Other parts include radio systems, mobile data terminals and automatic vehicle location amongst others. We are going to be referring to as the CAD specifications, the complete set of all system specifications including both hardware and software sub-systems. Whenever it is necessary to refer to the software specifications alone, we will do so explicitly.

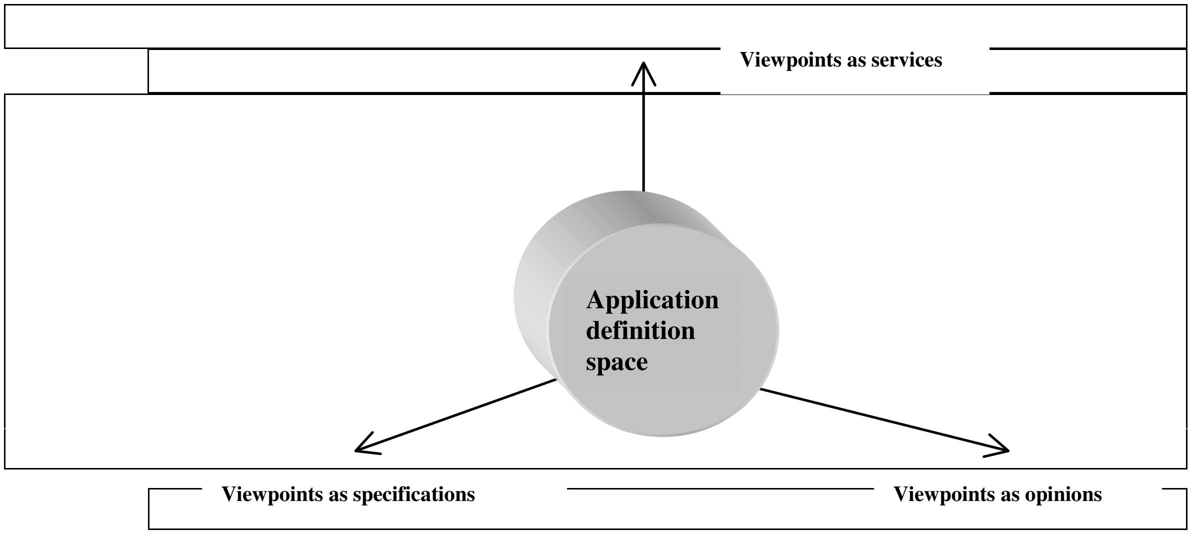

Figure2.1 – Viewpoints Classification according to [Leite96]

We conducted this study based on the viewpoint classification proposed by Leite in [Leite96]. In his framework, the author classifies current research effort in viewpoints into three orthogonal directions, viewpoints as opinions, services, and specifications. The classification is depicted in figure 2.1. We believe the use of the framework will help illustrate the scope of use of viewpoints. In the first section, viewpoints as opinions, we show that contrasting different perspectives, held by the stakeholders, can help bring to light potential conflicts. The following section, viewpoint as services, helps clarifying that the services provided by the system prove a good starting point to elicitation and modeling of its requirements. Finally, the last section, viewpoints as specifications, shows that integration among heterogeneous parts in complex systems can be facilitated by the use of viewpoints.

Viewpoints as Opinions

The construction of a large software system involves a great number of agents. These agents have different perspectives or views on the system-to-be that are directly related to the role the system will play in the future. These perspectives are partial or incomplete descriptions of the system and reflect the environment in which the system will operate. We believe that the integration of multiple perspectives can contribute to augment the overall understanding of the system.

The idea of using multiple sources of information goes back a long time. Criminal investigation uses this resource as a means to uncover hidden information and elicit details. However different testimonies may provide conflicting or inconsistent information on an issue. Inconsistencies, if properly used, can benefit a knowledge acquisition process [Easterbrook96]. Finding inconsistencies in specifications, specially those arising from conflicts between different parties, usually indicates that further knowledge elicitation is needed. Inconsistencies are, therefore, very useful to find mistakes, missing and mismatch information in requirements specifications.

In the case of the London Ambulance Service it is clear the presence of a multitude of different parties. Among the stakeholders, the roles of the system manager and ambulance staff clearly stand out. It is only fair to imagine that the requirements elicitation process for the system would take into account their points of view. Apparently that was not the case, as there is no evidence that the system took into account any opinion other than the system manager's, as stated bellow:

> *"the work was done primarily by the contract analyst with direct assistance from the Systems Manager… The proposed new system would impact quite significantly on the way in which staff carried out their jobs, yet in the case of the ambulance crews there was little consultation on this new method of working." [3016]*

The system would completely change the method of work of both staff and ambulance crews and yet there was little consultation regarding these changes. Senior management believed that the implementation of the system itself would be sufficient to bring about these changes.

Our claim is that the requirements engineering stage for the LAS provided an excellent opportunity for using an early elicitation strategy such as the one proposed by Leite and Freeman [Leite91]. Their method compares different views of a given situation, and partially supports the negotiation process necessary to reconcile different opinions. There are many advantages on finding conflicts this early in the development process, the main one is allowing validation during the elicitation of the system requirements. Conflicts of interest, such as the one described above, would be detected much earlier in the process, and perhaps some could have been circumvented or resolved through negotiation. As such, these problems only arose when the system was implemented.

A specific episode that would probably profit from the use of viewpoints is the unit allocation procedure. In this example, we find in the LAS report clear indications of the points-of-view of both management and ambulance crews. While the first understood he optimum choice to be the <u>nearest</u> to the incident, the station and ambulance crews used <u>other criteria</u> to allocate units to an incident. One of them was based in the proximity to the ambulance home base, so that the crews that operated in a better known territory and would not take too long to go back to base at the end of shift .

> *"The system would allocate the nearest available resource regardless of originating station, crews were having often to operate further and further from their home base. This resulted in them operating in unfamiliar territory with further to run to reach their home station at the end of a shift….The new system took away the flexibility they previously had for the station to decide on which resource to allocate." [3117]*

The current practice before the system installation was that the station or crews would decide which unit to allocate to an incident. This was not considered ideal by management and the new system would take up this responsibility and impartially allocate the optimum resource to any accident .

> *" A CAD system operating as it was intended in absolutely objective and impartial way would always ensure the mobilization of the optimum resource to any incident. This was perceived as to overcome many of the working practices seen by management to be outmoded and not in the best interests of providing the best service to London. Such practices included the previous ability of the crews themselves or the stations to decide which resource to mobilize to an incident." [3116]*

Shifting the ability to decide from men to machine was not a welcome change and made the system impersonal. The report suggest that the system was used in a way to "force" some changes to the existing practices management was not satisfied with.

> *"The lack of voice contact made the whole process more impersonal and exacerbated the "them and us" situation;" [3117-c]*
>
> *"CAD would eliminate those practices or so management thought."[3116]*

The outcome was that the staff found the changes to their work methodology imposed by the system as a "strait jacket" within which they still tried to operate local flexibility.

> *"Management and staff of LAS are supportive of the use of technology to enhance the service provided." [1007-b]*

It is clear that the management and the ambulance staff had opposite opinions in regard of the unit allocation procedure and the system was used to settle the matter. Conflict resolution between parties is a collaborative process and requires a great deal of negotiation [Easterbrook94, Boehm96]. In the case the LAS all evidence leads to believe that this process was overlooked as the system simply implemented using the management's viewpoint.

The consequences, as described by the report, were disastrous. The lack of joint ownership of the system by the ambulance staff resulted in an altogether negative inclination towards the system and its installations. The report describes the existence of circumstantial evidence of willful damage or misuse of the equipment.

Viewpoints as Services

We can regard the CAD system as a provider of services to the its users, namely the ambulance and ground staff. In this light, the computer aided dispatch system provides automatic support to tasks that were formerly performed manually. Among these tasks were call taking, resource identification, resource mobilization and ambulance resource management. A totally manual service presented some deficiencies, as the report clearly states:

> *The physical movement of paper forms around the Control Room was inefficient. [3006-b]*

> *Communicating with ambulances via voice is time consuming and, at peak times, can lead to mobilization queues.[3006-d]*

The implemented CAD system, needless to say, was expected to overcome these deficiencies. According to the report, the implementation brought some side effects:

> *The physical changes to the layout of the control room on 26 November 1992 meant that the CAC (central ambulance control) were working in unfamiliar positions, without paper backup, and were less able to work with colleagues with whom they had jointly solved problems before. [1001].*
>
> *The lack of voice contact made the whole process more impersonal and exacerbated the "them and us" situation;[3117-c]*

Kotonya and Sommerville propose a viewpoint-oriented requirements definition method (VORD) that takes into account the services the system provides to its potential user groups [Kotonya95]. In this light, the system development takes into account all the stakeholders involved and assigns a different viewpoint for each party. The method provides explicit support to elicitation (identification of relevant entities) by providing a initial set of viewpoint classes. It also prescribes a set of heuristics to guide the process of customizing the viewpoints to a particular system.

According to the authors, VORD can be applied to a wide spectrum of applications once it accepts notations varying from natural to formal languages. The idea is to enhance communication between users and developers by providing informal versions of the system specifications, allegedly more understandable to users.

We believe that modeling the system from the services provided to the staff viewpoint would have brought positive results. If not sufficient to foresee the effects the implementation would have on the former manual practices, as described above, it would at least contribute to establishing a ownership feeling towards the system.

Besides the services provided by system viewpoints, the technique proposed by Kotonya and Sommerville also address managerial and organizational aspects. According to the authors, software engineering processes are focused on user issues rather than organizational concerns, which leads to incomplete software requirements. In order to tackle this problem they introduced the notion of indirect viewpoints that incorporate organizational aspects related to the services delivered by the system, but do not interact directly with it. In this approach issues such as the influence the system may have on the organization and environment can also be addressed. They claim that indirect viewpoints are very important because people associated with them are often very influential in an organization and can make decisions on whether the system goes into service.

Ironically, the management's decision that the CAD system should go into service on the scheduled date, overriding both the user and developers viewpoints, is pointed by the Report of the Inquiry Team as one of the main contributions to the system's overall failure, as follows:

> *What is clear from the Inquiry Team's investigation is that neither the Computer Aided Dispatch (CAD) system itself, nor its users were ready for full implementation on 26 October 1992. The CAD software was not complete, not properly tuned, and not fully tested... it is difficult to understand why the final decision was made, knowing that there were so many potential imperfections in the system. [1001]*

Another way to address organizational issues is found in [Bowman96]. In this approach, the author introduces a set of viewpoints to separate system constraints arising from different areas of concern in the design process. The viewpoints enable participants to observe a system from different perspectives. There are five different (fixed) viewpoints presented in this framework, namely enterprise, information, computational, engineering and technology. This set of viewpoints was chosen so that, according to the authors, together they address the complete set of concerns involved in providing a specification of a large and complex system. One interesting aspect of this approach that could prove of special value to the CAD system is the fact that for each viewpoint there is a language associated. The enterprise and information languages, in special, are primarily intended to describe the environment in which the system is to be used and might have proven useful to prevent the situation described below:

> *The physical changes to the layout of the control room on 26 November 1992 meant that the CAC staff were working in unfamiliar positions, without paper backup, and were less able to work with colleagues with whom they had jointly solved problems before. [1001]*

The enterprise language proposed by Bowman introduces concepts to support the expression of *policy*, particularly with regard to agreements and responsibilities between parts of the enterprise. The fragment below seems to reinforce the importance of this piece of information.

> *…(recommendations from the Inquiry Team)…the next CAD system must be made to fit the Service's current or future organizational structure and agreed operational procedures. This was not the case with the current CAD. [1003]*

However, it is important to notice though, that the fact that the implementation of the CAD did not reflect the current organization structure and operational procedures was deliberate. The management wanted to use the system to change the current practices. Our point here is that the usage of viewpoints would perhaps make the organization goals more clear, perhaps facilitating their implementation.

Viewpoints as Specifications

The computer aided dispatch (CAD) system, it must be understood, is a combination of many components in which software is only a part. A CAD system provides one or more of the primary command and control functions of call taking, resource identification, i.e., determining which ambulance to send, resource mobilization and ambulance resource management in order to minimize response times. Depending on the functions to be performed, the CAD system consists of a combination of many different components. These components are, namely, CAD software, CAD hardware, gazetteer and mapping software, communication interface, radio interface system (RIFS), mobile data terminals and automatic vehicle location system (AVLS). It is only fair to suppose that the specifications for a system of this magnitude should be composed of a number of different fragments. These must comprise specifications written in various notations as the result of the use of a variety of methods. The fragments below should give a better idea of the complexity and variety of issues involved in the system specifications.

> *It is clear that in a system such as this where voice and data transmissions are so important it is vital that the communications infrastructure is beyond reproach. [3104]*

> *During the selection process it is worth noting that other bidders raised questions…doubts were raised as to the ability of the communications system to cope with the potential load to be placed upon it, and of the reliability and the state of readiness of the Radio Interface System (RIFS). [3058]*

> *Certain things such as Datatrak location fixing inconsistencies and communication failures can only happen in real time and cannot be easily simulated …a system such as this needs perfect information at all times. [3087]*

> *(in relation to human resources) it is important to understand that in any system implementation the people factor is as important, and arguably more important, than the technical infrastructure. [3107]*

> *Crew training and central ambulance control (CAC) is a key issue. [3112)*

This situation seems to be common place in engineering practice. Large systems are comprised of many parts of sub-systems, each specified using a different set of rules and standards of specification. In software practice this problem arises as systems grow larger and start using different notations and specification languages to address specific parts. In a way this approach helps clarifying the overall specifications of the system, because each part can be specified using the notation best suited for it (whether formal, object oriented, functional, etc..). On the other hand, it introduces some complexity regarding the integration of multiple parts.

A survey on some of the methods that tackle the complexity problem by using multiple viewpoints is presented in [Finkelstein93]. The problem consists of integrating perspectives written in different requirements languages that contain knowledge represented in various ways and may be in distinct stages of elaboration (taking in consideration that development might be carried out concurrently). Many methods are reviewed, approaches ranging from the use of classical logic to integrate assertions distributed in partial specifications to the use of common data models. All approaches are unanimous in stressing that separation of concerns is effective in reducing the complexity of large systems.

The integration of heterogeneous components in complex systems is addressed by Nuseibeh et al. in [Nuseibeh94]. The authors propose a framework in which to hold specification fragments that encapsulate partial representation knowledge, development process knowledge and specification knowledge about a system and its domain. Each fragment is a ViewPoint. The framework provides means to integrate those fragments into a larger, unique specification. It also helps guiding and organizing the software development process by means of providing specific templates in which to record different ViewPoints of the system. Figure 2 depicts a template.

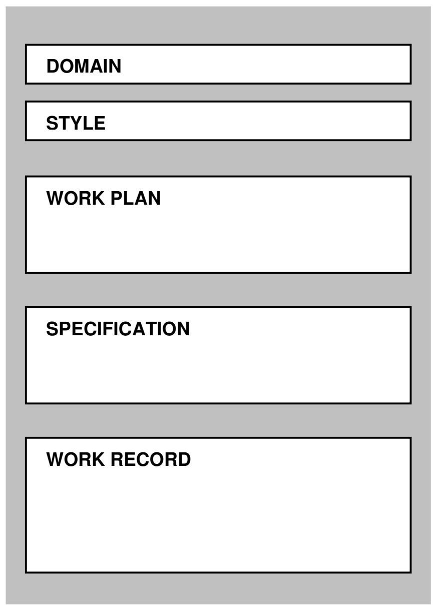

Figure 2 – ViewPoint template as proposed by [Nuseibeh94]

Notice that the templates are composed of many slots in which to define the domain of the application, the representation style, the specification itself and history and current state of development (work record slot). The work plan slot deserve special attention because it describes the process by which a specification in a style can be built. This aspect is very important because it incorporates development process knowledge to the system specifications, i.e., it makes it clear how the specifications were developed.

According to the authors the strength of the framework is the power to integrate different components of a large, heterogeneous system. It can express and check the relationships between specification fragments developed using multiple methods and notations. Positive aspects of this framework are the encapsulation of representation, development and specification knowledge data in each of the Views, thus facilitating local management and distribution.

In relation to the LAS, the ViewPoint approach is of special interest because it allows the accommodation of specifications in different stages of development, i.e., the system components do not have to be all specified at the same time and can be added to the specifications as the system evolves. In very complex systems it is only reasonable to suppose that the specifications are generated gradually, usually by distinct groups and using different notations.

Another approach to multiple specification integration can be found in [Berdon95]. The authors believe that no single notation is sufficient to capture all requirements of any given system. They propose a method that stores partial specifications of a system in a canonical representation, thus guaranteeing integration among different notations. Their definition of viewpoints precludes the existence of an owner, i.e., a viewpoint is a subset of all system requirements expressible in a given requirements notation regardless of the stakeholders involved. The idea is that no viewpoint is sufficient to describe requirements and the union of all viewpoints creates a “complete” set of requirements.

Central to their approach is the translation of any given specification into their canonical representation. Translating specifications from one language to another does not seem an easy task specially because some aspects that can be clearly stated in one language may not have counterparts in others. Nuseibeh points out that it is particularly challenging to describe relationships between different notations in a generic manner, more so if they use representation styles with different underlying data models or schemas [Nuseibeh94]. In the specific case of the CAD system, where so many different sub-systems are involved, e.g., mapping software, communications, radio, mobile data terminals, etc.., it is difficult to foresee a single representation powerful enough to express all the necessary requirements. We believe that it is more fruitful to concentrate our efforts in the integration of different notations then in designing yet another representation.

3. Evolution

Large-scale system development imposes a certain number of demands on their developers. One of them is that the requirements should be traceable in the face of the system’s evolution. It is a generally agreed fact that requirements change during the life cycle of a system, and keeping a good record of this changes is paramount for managing large-scale systems. The fragment below illustrates the traceability problem in the context of the development of the computer aided dispatch system(CAD) for the London Ambulance Service:

SO[2], in their eagerness to please users, often put through software changes "on the fly" thus circumventing the official Project Issue Report (PIR) procedures whereby all such changes should be controlled. These "on the fly" changes also reduced the effectiveness of the testing procedures as previously tested software would be amended without the knowledge of the project group. Such changes could, and did, introduce further bugs. [3082].

The practice illustrated above is a very common in industry [Pressman92]. Developers keep adding new features to their systems in a rather naïve fashion, disregarding the extra amount of effort it represents in testing and maintenance, the impact on previously defined requirements and the possibility of insertion of new bugs.

A recent empirical study in requirements traceability conducted by Gotel and Finkelstein [Gotel93] brought to light a series of important findings. Results from the study showed that many of the available commercial tools did not cover traceability properly, and that the quality of the requirements traceability obtained was directly related to the adherence of the tool to a particular method or notation that it implemented. The most noteworthy finding however, was that traceability problems could be divided into two main categories, those related to aspects of a requirement's life prior to the inclusion in the Requirements Specification document (pre-RS), and those related to aspects of a requirement's life that result from the inclusion in the requirements specification document (post-RS). Figure 3.1 illustrates the separation. Conclusive information was presented showing that pre-RS traceability was neither well understood nor fully supported. It also suggested that if no special effort is conducted to aid the capturing and recording pre-RS data, it is likely that this information will be lost.

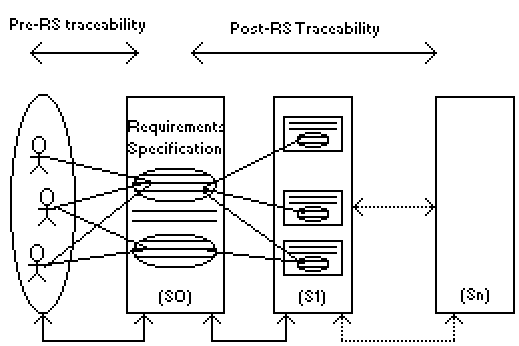

Figure 3.1. Two basic types of requirements traceability

[2] SO stands for Systems Options Ltd., the software house that won the bid.

We dedicate this section to looking deeper into the requirements traceability problem through the prism of our case study, the CAD system. We must, however, note that the majority of problems related to traceability arise during system maintenance [Leite 95] and the CAD system was put out of service before reaching this stage. Our effort is thus concentrated in the capture of useful information during the development process that could be of help in tracing the requirements back to their origins. We aim to identify opportunities where some of the methods and techniques reviewed could be useful in bringing to light issues such as poor management, and lack of control of changes (illustrated by the fragment above) that ultimately contributed to the overall failing of the system and its premature discontinuation.

Traceability and the software development process

Developing large-scale software systems is a very complex activity. It involves a great deal of people and resources that have be coordinated towards the same goals. Organization can be achieved in a number of ways, one of them is producing various models of the system and gradually refine them. By introducing system complexity in successive models, it is possible to manage system complexity [Jacobson94]. In this light, the relationships between models play an important role, i.e., objects in one model must be traced to objects in another. This concern is reflected in a number of standards, such as the IEEE [IEEE-Std830-1984] and the SEI [SEI-CM1990].

Concern with the traceability aspects of system development have been present in the literature for quite a long time, and can be traced back to the Software Requirements Engineering Method (SREM) [Leite87]. Many commercial tools and research products support traceability at present, implementing a wide spectrum of techniques that range from the use of templates to traceability matrices. Gotel and Finkelstein present a wide comparative study over 100 tools that implement traceability to some degree in [Gotel93].

To the best of our knowledge, the only tool used to perform traceability during the development of the CAD system was the PIR system that implemented the change log. At one point, the Report on the Inquiry of the LAS mentions that changes were being made to the system without the control prescribed by the Project Issue Report (PIR) procedures, i.e., changes were being made but not recorded. There is evidence that this practice continued on to the implementation stage of the system, as follows:

> *…Many of these problems stemmed from the fact that printers being used in this way were never part of the original specifications, but were added in a haste as a short term expedient to show some positive progress at an already published implementation date [3088].*

Intimately linked to configuration management and versioning aspects, traceability concerns have also emerged from that perspective [Bersoff80]. These activities, usually called "umbrella" because they permeate the entire software life cycle, have to be organized in a manner that enables the orderly control of change [Pressman92]. Configuration management aspects are specifically important to the CAD system as the implementation of the system has been done in pieces, across different divisions of the LAS. During that period, so many changes were made to the system that its stability was permanently at stake. It seems that the developers were also loosing track of the changes made, as an independent reviewer recommended that the change log should be under more formal controls. The fragments below illustrate this scenario:

> *Over the first 9 months the system was implemented piecemeal across different LAS divisions in the following phases (although there were variations within each phase) [3093]:*

> *During these months the system was never stable. Changes and enhancements were made continually to the CAD software. The Datatrak system was being similarly amended and enhanced. The MDT's and the RIF system were also undergoing continuous changes [3098].*

> *In March 1992 the LAS system manager was asked to carry out a further independent review of the project progress. In his report he makes the following observations and recommendations:*
>
> *d) the log change (PIR system) should be under the formal control of the Systems Team and CAC should not be allowed to agree piecemeal changes with SO outside of this formal system; [3100]*

It is clear from the last fragment that the reviewer feared that changes were informally being agreed to by the developers and users. We, by no means are claiming that these changes should not be made. We believe that pleasing the users is good Software Engineering practice and should be among the goals of any practitioner. Our point here is the necessity of providing means, preferably automated, to record and control change.

Borrowing a concept from configuration management, Leite offers support to traceability throughout development, by providing a perennial requirements baseline [Leite95]. The baseline runs parallel to system development, and performs both pre and post-RS traceability. The baseline is comprised of a Hypertext, a Basic Model, a Configuration Views and was recently extended to incorporate a Lexicon and a Scenario Model views [Leite97]. The Basic Model View is depicted in figure 3.2.

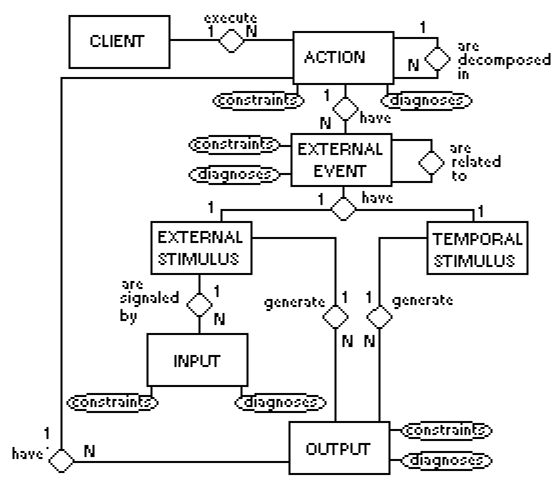

Figure 3.2. The ER Diagram for the Baseline Basic Model View

The Basic Model View uses the entity relationship framework as a representation language. The basic components are clients, actions, external events, inputs, outputs, restrictions and diagnoses. The Configuration View is responsible for the versioning system behind the Basic Model. It makes explicit the person responsible for the changes and the rationale behind it. The underlying idea is the one proposed in the Contribution Structures Model [Gotel95]. In this model, the requirements artifacts are viewed as contributions and the agents involved in the productions are denominated contributors. Contributions and contributors are connected by the Contributions Relations, that register the level of participation of each contributor. The aspect of ownership is fundamental for both pre and post-RS traceability. A good account of modeling and use of contribution structures in practice can be found in [Gotel97].

The Hypertext view is composed of the relationships depicted in the Basic Model using a controlled vocabulary. The vocabulary is the representation of the symbols in the problem domain language and it is specified in the Lexicon View. Finally the Scenario View is responsible for introducing notions such as goals and making explicit relationships among resources and actors. The Scenario View also allows for the representation of parallel events and exceptions.

There is very little doubt that the LAS system could have benefited from the use of the requirements baseline approach. We believe that the Scenario View, in special, could be used to simulate exceptional behavior and evaluate the consequences of part malfunction. The CAD system was specified to rely on nearly perfect information but, as the implementation took place, a great many problems with the equipment came up. It seems the equipment malfunction were treated as independent episodes, never considering the impact they might have on the overall system. The fragments below illustrate this situation:

> *… its success (the system's) would depend on the near 100% accuracy and reliability of the technology in its totality. Anything less could result in serious disrupt to LAS operations. [3024]*

> Data transmission problems*: there were continuing problems with data transmission, many of which are still not totally resolved. [3079].*
>
> *Continued difficulties through mobile data through failures to transmit or receive signals and, occasionally, through MDT(mobile data terminal) lock up [3090].*
>
> Installation problems*: there were also instances of poor quality equipment installation by Datatrak subcontractors [3080].*
>
> Problems caused by mistakes*: an error in a formula provided by Datatrak to SO was not clear until October 1992 [3090];*

Finally, another aspect that must be addressed is the volume of information involved in performing system traceability. Tracing requirements involves very large amounts of data, making tool support a must. According to Leite "*without automation traceability is a myth*" [Leite 95].

A crucial issue in requirements traceability is, therefore, selecting the relevant information that is to be kept from the less meaningful data that should be discarded. The objective is to keep the volume of traceable information in a tractable level. This issue is addressed by Pinheiro and Goguen in [Pinheiro96], where the authors propose a tool that performs contextualized tracing of information. The authors understand that requirements are situated, i.e., they depend upon details of the particular situation and context in which they arise, and implemented a tool that uses this aspect to perform selective tracing. This issue is also addressed in the PRO-ART environment [Pohl96]. The author proposes a three–dimensional framework for requirements engineering that defines the basic types of information to be recorded

It is impossible for us, with the documentation at disposal, to make an assessment of what information would be relevant to the traceability of CAD system. The Report of the Inquiry hardly provides any technical details and any attempt from our part would result in a misleading reduction of what the actual problem is. Nevertheless, we want to list some of the problems that strike us as most relevant to the situation. The first one is keeping track of changes made to the various interfaces to the hardware pieces of system. We often see reference to software requirement traceability but very little is said about maintaining consistency across different hardware pieces. Secondly we believe that a larger attempt to formalize the environment in which the system is to operate would be of much help to future enhancements. Of course this is to say that we must consider the London Ambulance Service from a society at large point of view, therefore taking into account issues such as laws, policies and government legislation.

Pre-RS traceability

Traceability is defined as the ability to describe and follow the life of a requirement in both forwards and backwards direction [Gotel93]. The backwards direction corresponds to the pre-RS requirements, i.e., those related to the requirement's life prior to its inclusion in the RS document. Pre-RS traceability depends on the ability to trace requirements from, and back to, their originating statements, through the process of requirements production and refinement, in which statements from diverse sources are eventually integrated into a single requirement in the RS. It obvious that understanding the nature of the stakeholders contributions in the requirements process play a major role in the supporting traceability [Gotel95]. In the case of the development of the CAD system, the team responsible for the elaboration of the system requirements was composed of representatives from management alone. Neither representative from the ambulance staff and control were present, as the fragment below states:

> *In order to prepare the requirements specification for the proposed new system, a team was assembled under the chairmanship of the Director of Support Services with the then Systems Manager, a contract analyst, and the Control room Services manager… Because of problems at the time with the staff consultation process there was little involvement at this stage from the ambulance crews, although invitations to participate were given to union representatives [3011].*

There is little doubt that representatives from the ambulance staffs and control room would have contributed to augmenting the understanding of the system at the requirement stage and help elaborate the software requirements specification (RS) document. In this situation it is only fair to conclude that the people that elaborated the requirements for the CAD system represented one, and only one perspective, that of the management. In this case, we can safely conclude that the available pre-RS traceability information for the system, if any, is totally biased towards the managerial view of the system. This conclusion, it must be said, is consistent with the goal of using the CAD system to change inadequate working practices manifested by the management, as follows:

> *Management were misguided or naïve in believing that computer systems in themselves could bring about such changes in human practices [3116]*

As mentioned before, since the CAD never got into the maintenance stage, there was very little opportunity in which the emergence of traceability related problems could be observed. However, because of the biased nature in which the RS was elaborated, clear manifestations of disregard for the system were noticed during its development and implementation, such as the evidence below:

> *There is little doubt that perceived and actual problems with the system did cause resentment amongst some ambulance crews which would have affected their attitude to the system. [3117]*

> *There is no confidence in the with the result that the staff expected it to fail rather than willing it to succeed.[3108]*

According to [Gotel93], the lack of pre-RS accounts for the majority of problems that result from poor requirements traceability. In order to overcome some of the problems encountered during the pre-RS traceability stage, the author identifies some aspects that deserve more attention, namely, increasing awareness of information; obtaining and recording information; organizing and maintaining information; access and presentation of information.

In relation to the obtaining and recording information aspect, Wood in [Wood 94] utilizes multimedia support to requirements capture. The main argument of this approach is that through the use of multimedia one can store the requirements in a format very close to their original. In general, when rough data is transformed into more formal specifications, relevant information is lost and the traceability of the original information may be more difficult. The use of multimedia support can minimize these problems and help reduce requirements volatility. In the case of the LAS we believe that multimedia facilities would be of special interest in capturing requirements related to the workings of the control room. The former manual operation is very complex involving a variety of human actions. The fragment below describes the call taking of the operation. Both the resource mobilization and identification procedures are fully described in the report of the inquiry.

> *When a 999 or urgent call is received in the Central Ambulance Control(CAC) the Control Assistant(CA) writes down the call details on a pre-printed form (AS1 or AS2). The incident location is identified from a map book, together with the map reference coordinates. On completion of the call the incident form is placed into a conveyor belt system with other forms from fellow CA's. The conveyor belt then transports the forms to a central collection point within the CAC. [3002]*

Notice that description above is very simple. No exceptional behavior is described, neither the heuristics applied to choose between the two different forms. Understanding human practices is a very complex task because it involves a great deal of tacit knowledge, that is very difficult to formalize [Goguen94]. We believe that in this specific case the use of multimedia, video specifically, could come as a solution to capturing the working practices in the Central Ambulance Control . Jirotka demonstrates in [Jirotka95] that the use of video can support the requirements process, in specific how fieldwork augmented by video recordings can help identify important features for requirements practice.

Another approach to recording the origins of a requirement is capturing the rationale involved in its elaboration. Potts and Bruns propose an approach based on the IBIS model that captures design rationale by recording discussions and decisions about requirements in a network of *issues, positions* and *arguments* [Potts88]. In the case of the LAS, it is a generally agreed fact that the CAD system was implemented against an impossible timetable. What is not clear is the process by which the figures were established. What estimates were used and the arguments in favor or against those. Understanding the rationale behind these calculations would certainly have been beneficial to the management of the project as a whole, as it would allow for "on the fly" adjustments on the deadline requirements. The fragment below illustrates this scenario:

> *The lack of consultation and agreement over realistic deadlines with managers and staff involved in the process, and the pressure imposed by going public on deadlines and the then finding difficulties in meeting them, clearly needs to be avoided in the coming years. [6022]*

Another hypertext-based approach is the one implemented by the RETH tool [Kaidl93]. This approach supports traceability through the use of hypertext, that serves as a mediating representation between informal ideas of the user in the very beginning and more formal representations.

Goal oriented approaches, such as the ones proposed by [Antón96, van Lamsweerde95], are probably also beneficial from a traceability perspective, for they aim to capture goals and their relationships to agents, actions, constraints and other objects in the environment. The report on the inquiry of the LAS system presents a comprehensible list of goals for any future CAD besides providing the basic command and control functions of call taking, resource identification and allocation and ambulance resource management [IR–LAS 2014]. We exemplify some of them, as follows:

> *The new system must contribute to improving the level and quality of the provision of ambulance services in the capital;*
>
> *It must be fully reliable and resilient with fully tested levels of back-up;*
>
> *It must have total ownership by management and staff;*
>
> *Management and staff must have total, demonstrable, confidence in the reliability of the system; [5004]*

It is true that traceability will arise as a consequence of the use of the approaches discussed so far, for all aim at capturing requirements in general and not performing traceability in specific. According to Leite, requirements traceability is a process that must occur in parallel to system development, supported by its own tools [Leite97].

Pohl proposes a requirements engineering environment aimed specifically at capturing information that enables pre-RS traceability. The Process and Repository based Approach for Requirements Traceability (PRO-ART) provides framework in which to record the information, a repository in which to store the information and a tool that allows (almost) automatic capture of traceable information. The framework in which to capture the information is three dimensional; it comprises specification, representation and agreement dimensions. The specification dimension ranges from opaque to complete and measures the level of understanding of the system requirements. The representation dimension ranges from informal to formal and measures the level of formality in the specifications and, finally the agreement dimension ranges from personal to common views and measures the level of integration of the stakeholders viewpoints on the system [Pohl96]. This framework symbolizes the author's understanding of requirements engineering prior to the elaboration of the RS, i.e., the time frame in which pre-RS traceability takes place.

4. Environmental Aspects

Another dimension of the development of the Computer aided dispatch (CAD) system for the London Ambulance Service (LAS) is the influence the social structure of the organization had on the development and deployment of the system. It is a known fact that software development does not happen in a vacuum, i.e., it is impossible not to suffer bias from the surrounding environment and organizational structure. In this light, the acquisition of a better understanding of the Universe of Discourse in which the application will be deployed is paramount to the success of development [Davis82]. By domain we understand the part of the world in which the computer system effects will be felt, including its people, organizational structure, related legislation, physical locations and not only the computer system.

Understanding the domain in which the CAD system was to be imbedded is a very complex task. For one thing the London Ambulance service provides a public service to the community at large and is controlled by the not less complex South Thames Regional Health Authority, that in turn is accountable to the National Health Service Management. However interesting a discussion on the role of LAS in the society, it is outside the scope of this report. We will concentrate our efforts in the identification of specific problems highlighted in the report of the inquiry, and try to demonstrate that these problems could have been identified and perhaps avoided by the use of a philosophy or software process that aims at tackling social aspects impact on the software development process and deployment.

According to Kramer and Wolf the "root causes of failure of the LAS project appeared to be organizational" [Kramer96]. Finkelstein point out the "poor fit of the system with the organizational structure of the ambulance service" and the "lack of consultation with clients and users in the development process with knock-on consequences for their ownership of the resulting system" as some of the problems identified by the inquiry. The passages bellow evidence these facts:

> *It is worth noting that in general the attitude to the use of information technology amongst the CAC management and staff, and the ambulance crews, is overwhelmingly positive. All recognize that computerization of this area is essential in enhancing efficiency. However, there is an equally unanimous feeling that current CAD system, and particularly the way in which it was introduced, is not the way ahead. There is no confidence in the system with the result that staff expected it to fail rather than willing to succeed. [3107].*

> *The proposed new system would impact quite significantly on the way in which staff carried out their jobs, yet in the case of the ambulance crews, there was little consultation on this new method of working [3016]*

Indeed, during the requirements stage of the project there was very little user participation. This strategy may have contributed to the lack of confidence and ownership of the system demonstrated by great part of the staff. The impact those had on the deployed system is diversified and range from willful misuse of physical equipment (sabotage is hinted) to the caretaking of vehicles. In this section we will try to sketch this scenario, providing samples of the variety of the problems identified, at the same time that we point out to techniques that aim at the identification of the social issues directly concerning the computer system. It is important to notice that none of the techniques that will be surveyed propose solutions to any to problems identified in the social context, they merely identify and try to foresee possible impacts those may have in the system development and deployment. The perspective here is that of an outside observer, that records events but does not try interfering with them. In this light, the idea of taking in consideration the social environment in the realm of Requirements Engineering is that of making a better planning of the system and augmenting its chances of success. The possibility of solving identified problems in the social context is not even considered, for that demands supporting the hypothesis that a software system by itself would be able to work changes to its environment. We know that not be true, although it is reported that the inclusion of some systems did change its environment [Lehman80], but those did not result from the system presence alone but counted with cooperation from people and the organization surrounding it.

Ironically enough it is reported that the LAS tried using the CAD system to change some of the current working practices management was dissatisfied with. The changes were to be implemented by the system alone, without collaboration from ambulance crews, ground staff or the management itself.

We are going to use the unit allocation procedure to illustrate this discussion. The current practice before the system installation was that the station or crews would decide which unit to allocate to an incident. This was not considered ideal by management and the new system would take up this responsibility and impartially allocate the optimum resource to any accident. Shifting the ability to decide from men to machine was not a welcome change and made the system impersonal. The report suggest that the system was used in a way to force some changes to the existing practices management was not satisfied with. The following passages illustrate this issue:

> *A CAD system operating as it was intend in an absolutely objective and impartial way would always ensure the mobilization of the optimum resource to any incident. This was perceived to overcome many of the working practices seen by management to be outmoded and not in the best interests of proving the best service to London. Such practices included the previous ability of the crews themselves or the stations to decide which resource to mobilize to an incident. CAD would eliminate this practice, or so management thought. Unfortunately such practices could not so easily be eliminated and the CAD system would accommodate them only with difficulty and with a reduced level of efficiency. Crews and stations could, if they wished, still accommodate these practices by failing to mobilize, or sending a different resource or failing to acknowledge or report status. Such practices would contribute to clogging up the system. Management were misguided or naive in believing that computer systems in themselves could bring about such changes in human practices. Experience in many different environments proves that computer systems cannot influence change in this way. They can only assist in the process and any attempt to force change through the introduction of a system with the characteristics of an operational "straitjacket" would potentially be doomed to failure[3116].*

> *"CAD would eliminate those practices or so management thought."[3116]*

> *The outcome was that the staff found the changes to their work methodology imposed by the system as a "strait jacket" within which they still tried to operate local flexibility. [3116]*

It is clear that the management and the ambulance staff had opposite opinions in regard of the unit allocation procedure and the system was used to settle the matter. Only a great understanding of the organization structure, its people and working practices could have brought to light this issue during system development. Obtaining this knowledge is no easy task. A possibility is to make use of ethnography to support the knowledge acquisition process. Ethnography is a study within the Sociology realm and is concerned with the description of social lives and of the activities occurring within this context.

Sommerville in [Sommerville93] and Randall [Randall93] present examples of the application of ethnography in the requirements elicitation process. An applied ethnographic study during requirements elicitation would help understanding and identifying current working practices. The insights resulting from this study can point out to cooperation aspects of the new system. Another contribution is the perception of tacit knowledge present in the execution of any activity. In general, traditional elicitation techniques such as questionnaires and interviews, are not capable of identifying tacit knowledge [Goguen94].

Another success story in the application of an ethnography based approach to a real case can be found in [Luff 93, Luff 94] where the author describes the necessity of requirements elicitation process that takes into account the social context and existing collaborations. The author arguments against traditional techniques that are biased by the person doing the elicitation. In order to overcome these difficulties Luff applied ethnography techniques allied to video taping. The video made possible to repeatedly analyze a task. It is claimed that through this approach it is possible to examine how the tasks were done in loco, for one is suppose to expect bias from techniques that elicited information from users outside their place of work, such as protocol analysis [Lingaard94] . Other advantages are perceiving how apparently individual tasks are done interactively with other members of the staff and analyzing the collaboration among members of a group.

However there are some drawbacks to the application of ethnography based techniques during requirements elicitation. Firstly there is the cost, an ethnographical study can take up a long time and consume a great deal of resources, thus increasing its overall cost. Secondly the ethnographical format of the resulted information is not fitted to be integrated the requirements documentation directly. The discourse, notation and format of presenting data used by ethnographs is very different from the format information is presented in a requirements specification document. Finally there is the system focus, an ethnographical study will provide information on the current state of affairs only. An ethnographical study is not predictive, therefore it will not provide speculations on possible consequences the introduction of a new system may have on the studied environment.

Goguen, in order to tackle some of these drawbacks, recommends a zooming technique for system requirements elicitation [Goguen94c]. In this approach the author claim that one should first apply tradition requirements elicitation techniques to make an overall model of the problem at hand. Some requirement elicitation techniques are reviewed in [Goguen94b]. On a second stage more costly techniques are to be used selectively to solve problems previously identified by other techniques. The method is then comprised of first making a general study that aims to cover the basic aspects of the social order, i.e., categorization system used by members, division and goals of social groups. This task can be accomplish with the aid of questionnaires and interviews. On the second stage a more elaborate ethnographical study is to take place in order to explore problems understood as critical to members. Finally the author suggests that discourse or interaction analysis should be applied to deepen specific aspects of the problems.

Another alternative to the problem was proposed by Leite [Leite83]. The author proposes a working method for systems definition through greater user participation and better understanding of the social impacts of the system to be. The method comprises three stages. The first is the investigation of the surroundings of the system. In this stage one is to acquire a better understanding of the Universe of Discourse the system is embedded in, i.e., the environment in which the system is going to be deployed. This task included recognizing and identifying social groups connected to the system. On the second stage, a committee is to be created in order to promote system discussion among different social groups. Finally a study of the system possible impacts is made. In order to accomplish this task it is necessary to apply an analysis of the social dimensions of problem. One possibility to perform the social dimension analysis is the framework proposed by Kling [Kling82]. Among the dimensions listed by author there are: ideology, power structure, work environment, infrastructure, quality of services provided and organization culture. A system as complex as LAS, that involves a great number of different user groups, provides a excellent object for a deeper socio-technical analysis.

Any of the techniques discussed above would aid in bringing to light some of the social aspects related to the development of the CAD system for the London Ambulance Service. The most noteworthy of these issues, that of the working practice was already discussed. We dedicate the rest of this section to illustrating minor issues that although subsidiary to the point are nevertheless important.

4.1 Impact of the chosen technology on the staff and ambulance crews

The inquiry team strongly believed and showed accordance to the technology solution proposed. In particular there is no trace of a discussion whether a totally automated system was the ideal solution in terms of social aspects. For instance, the substitution of voice contact was never analyzed as a non desirable side effect. The report on the inquiry mentions the fact that information technologies that replace voice contact degrade the social environment. The passages bellow illustrate this point:

> *"these and many other potential advantages make a convincing case for automation" [3008]*
>
> *"…there was unanimous support for technology to be used to enhance the delivery of ambulance services in London." [3009]*
>
> *"The lack of voice contact made the whole process more impersonal and exacerbated the "them and us" situation;" (3117-c)*
>
> *"There is much circumstantial evidence of willful damage or misuse of Datatrak equipment." [3110]*

The issue of technology dissemination in organizations is beside the point of this paper. An interesting discussing on the adoption of new technologies in a large governmental organization can be found in [Walz93, Basili91, Kellner91, Pinto93]. We believe that other factors were involved in the apparent rejection of the automated system by both ambulance crews and control room staff. Those facts are intimately connected to the consultation and ownership of the system by the same people. As remarked by Tom DeMarco, sociological issues such as team harmony, system ownership and group sense of purpose can impact development more then technological issues [DeMarco91]. The passages bellow corroborate this point:

> *"The proposed new system would impact quite significantly on the way in which staff carried out their jobs, yet in the case of the ambulance crews, there was little consultation on this new method of working." [3016]*

> *"A key feature of the failure of the CAD system is the inability of CAC and ambulance staff to fully appreciate each other's role in providing a service to London." [3113]*

4.2 Physical disposition

Along with the system deployment there were a series of minor adjustments to the working environments of both the ground staff and the ambulance crews. Those changes were necessary to support the installation of the technological devices that were part of the computer aided dispatch system. They included the installation of the MDT's on the ambulances and making changes to the disposition of the control room, among others. There is no evidence that these were take into account during system development much less user consultation on the matter. The passages bellow illustrate the situation:

> *The physical changes to the layout of the control room on 26 October 1992 meant that CAC staff were working in unfamiliar positions, without paper backup, and were less able to work with colleagues with whom they had jointly solved problems before. [1001].*

It seems this discussion borders the frontiers with ergonomics. To the best of our knowledge, none of the surveyed techniques take specific interest in the disposition of working environments or possible changes to them. What is the realm of requirements engineering is foreseeing the impact possible changes might have in the success of the system. That includes overall usability and ownership by the users. We believe that it is reasonable to say that some requirements engineering techniques, such as the ones proposed by [Goguen94] and [Sommerville95], address the ergonomic problem to some extent. Seen in a prudential light, we can say that by incorporating issues such as organization structure and its universe of information one is indirectly tackling the issues of organizing the working space. Some of this issues are addressed from a requirements perspective in [Luff94].

4.3 Training

The lack of proper training of both ambulance crews and ground staff comes as a side effect that contributed to the undermining of the overall confidence on the CAD system. There is evidence that training was provided much ahead of time and by the time the teams were to work with the system the "skill decay" effect was observed in some cases. It is not fair to imagine that the training issue alone played a major part in the system downfall. Again, it was included in this section as an illustration of the variety of side issues that contributed to general failure of the system. The passages bellow illustrate this issue:

> *"There was also doubts over the quality of training provided…..the training was not always comprehensive and was often inconsistent." [3112]*

> *It would be also necessary to have absolute confidence in the ability of the staff to work with the system. CAC staff would have to be fully trained in, and familiar with, the use of the facilities. Most importantly as the new system would change the operational method of working at the ambulance crew level it was particularly necessary to have their full co-operation and joint ownership of the system. If the crews did not press the right buttons at the right time and in the sequence systems chaos could result. [3026].*

> *"Central Ambulance Control and crews had no confidence in the system and were not all fully trained" [4001]*

4.4 Other issues

A plethora of different issues contributed to various extents to the overall downfall of the CAD system. Among these are social pressure to improve the quality of the services provided, an illustrated bellow:

> *LAS management were under constant pressure to improve performance and to meet the ORCON standards. LAS satisfactory implementation of the system would require changes to a number of existing working practices. Senior management believed that implementation of the system would, in itself, bring about these changes. In fact many staff found it to be an operational "strait jacket" within which they still tried to operate local flexibility. This caused further confusion within the system. [1007].*

Another interesting issue present in the report of the inquiry concerned the presentation of the ambulance crews and the vehicles themselves.

> *We recommend the reinstatement of recognized uniforms to all levels of operational and control management, including, in appropriate circumstances, senior executive staff. [6016]*

> *"To enhance the professionalism of the service as a whole, and to ensure that qualified ambulance staff are always in a position to respond 999(emergency) calls, we recommend that consideration be given to the employment of vehicle cleaners at stations." [6045]*

Large system development such as the CAD system involves a great number of complex social objects, restricted by their contexts, infrastructure and history. One conclusion that may be derived from this brief exposition is that no single factor can be actually pointed out as the major cause of failure of the system, but a very complex web of smaller issues. This fact seriously impacts the work of requirements engineers and supports the claim for new methods, tools and techniques that support social aspects of computer system development.

5. Non-Functional Aspects

In previous sections of this report we have mostly dealt with problems that have arisen as a result of technical difficulties or miscalculation by the system developers. In this section we will tackle problems of a more abstract nature, those related to the overall quality and attributes to be exhibited by the resulting system. These aspects relate to issues other than the actual functionality of the computer system but are, nevertheless equally as important.

Understanding the environment in which the computer system will be part of is very important to assure a successful implementation [Zave97]. Actually, the mere presence of some types of system in an organization will determine complete changes in its environment [Lehman80]. Getting a good grasp of the interactions between the computer system and its future environment is thus fundamental to good system development practice. It is clear that this scenario will bring to light new requirements, typically the ones related to the user perception of the system. These include performance, cost, time and operation constraints and are usually referred to as non-functional requirements.

The role non-functional requirements played in requirements engineering was first brought to light by Yeh in [Yeh84]. In this landmark paper the authors stress the importance of issues that are not directly related to the functionality of the system. Among these are its performance, maintenance, evolution and security related aspects. The authors also recognize the economical context of system development as a restriction to its requirements. In this light the authors see system development from an outside-in perspective, that is to take in consideration that large software systems are impacted by intrinsical changes to the outside environment. Therefore, changes to the system are inevitable and developers must be prepared to cope with them.

Reference:	Domain	Applicability	Non-Functional Requirements Classification	Non-Functional Requirements Sub-Classification
[Boehm78]	All systems	Product	Utility	Reliability: completeness, accuracy, consistency Efficiency: device efficiency, accessibility Human Engineering: accessibility, communicativeness
			Portability	Device independence Completeness
			Maintainability	Testability: accessibility, communicativeness, structuredness, self-descriptiveness Understandability: consistency, structuredness, self-descriptiveness, conciseness, legibility Modifiability: augmnetability
[Yeh84]	All systems	Product/ Process	I. Target System constraints	Performance : real time, other time constraints, resource utilization, accuracy, Reliability: availability, integrity
				Security: Physical, Operational
				Operating constraints: Frequency and duration of use, control, physical constraints,
			II. System Development, Evolution and Maintenance	Kind of development
				Scale of effort
				Methodology: quality control standards, milestones and review procedures, acceptance criteria
				Priority and changeability: relative importance of requirements, identification of factors likely to change
				Maintenance: Software: responsibility for fixing bugs, instrumentation Hardware: frequency and duration of preventive maintenance, responsibility for repair of faults, test equipment and procedures
			III. Economic Context of System Development	Cost Tradeoffs: utilization of existing technology versus development of new hardware or software, primary objectives: design-to-cost versus design-to-function
				Cost of iterative system development
				Cost of each instance of target system
[Mylopoulos92]	Information Systems	Process	I. Product oriented requirements	
			II. Process oriented requirements	
[McCall94]	All Systems	Product	Product Operations	Correctness: traceability, consistency, completeness Reliability: error tolerance, consistency, accuracy, simplicity, Efficiency: execution efficiency, storage efficiency Integrity: access control, access audit Usability: operability, training, communicativeness, input/output volume, input/output rate
			Product	Maintainability: consistency, simplicity, conciseness modularity, self-descriptiveness Flexibility: modularity, generality, expandability, self-descriptiveness Testability: modularity, simplicity, instrumentation, self-descriptiveness
			Product transition	Portability: modularity, self-descriptiveness, machine independence, software system independence Reusability: generality, modularity, software system independence, machine independence, self-descriptiveness Interoperability: modularity, communications commonality, data commonality
[Kirner96]	Real Time Systems	Product/ Process	Timing	
			Performance	
			Safety	
			Maintainability	
			Security	
			Usability	

Table I – Proposed classifications for organizing non-functional requirements.

In respect to the London Ambulance Service (LAS) system we are surprised to verify that non-functional requirements played a very extensive role in the downfall of the system and reference to those permeate the available documentation. Kramer and Wolf point out that the root causes of failure of the system lie somewhere in the boundary between organizational and technical issues [Kramer96]; Finkelstein identifies poor fit of the system with the organizational structure and undesirable performance among others, as some of the problems present in inquiry [Finkelstein96], and the IWSSD architecture group lead by Daniel Jackson reported that relevant aspects of the problem were that it was safety-critical, provided a time-critical service and should be usable [Kramer96].

In literature we have found many classification schemes for non-functional requirements. Table I exemplifies their variety that includes from product-oriented software quality models to specific work done in the context of requirements engineering.

This section is organized as to evidence some of the aspects present in the report of the inquiry on the LAS that qualify as non-functional requirements related issues. Among these are performance, integrity, user interface, etc…

5.1 Reliability

Perhaps the most important aspect concerning safety-critical system is the extent to what its users can confide in it, in other words, its reliability. It is important that users have a good perception of system actions and outputs, thus establishing a relationship of trust between them.

In the case of the LAS we will discuss system reliability using the resource allocation routine as our example. In the manual system the ambulance central would communicate to the ambulance via radio and allocate what they considered the best resource to an accident[*]. The CAD system substituted this practice by a completely automatic process that resulted from a combination of an automatic vehicle location system (AVLS), mobile data terminals, a radio system, a communication interface, gazetteer and mapping software and computer aided dispatch hardware and software. For this solution to work properly it was necessary that the system kept track of all ambulances at all times. This was to be done by the AVLS system but, unfortunately, in a communication environment as crowded as the city of London this was not always possible. The passages below exemplify the need for perfect

[*] A lot of issues were involved in deciding what the best resource, vehicle, was. Those are discussed in detail in following sections but omitted here on purpose for they are outside the point of this discussion. These issues were related to changes of working practices rather then having a direct involvement in the system overall reliability

information and some situations were lack of the precise vehicle information happened:

> *The system relied on nearly perfect information of vehicle location and status. Without accurate knowledge of vehicle location and status the system could not allocate the optimum resource to an incident. [4008]*

> *There were many imperfections in this information which individually may not be serious, but which cumulatively were to lead to system "failure". [4017]*

> *(Possible reason for the dispatch system not knowing the correct vehicle location) Failure of the system to catch all the data [4009-a]*

> *(Possible reason for the dispatch system not knowing the correct vehicle location) Poor coverage of the radio system, i.e., black spots. [4009-c]*

We must understand that the system was naively conceived to operate in this condition, i.e., having perfect information on all vehicle location at all times. It is not reasonable to expect that the delivered system operated in those conditions. A series of errors attributed to lack of vehicle information happened and resulted in the loss of reliability the users had on the system, as shown in the passages below:

> *The occasional imperfections of the Datatrak reduced confidence on the system [3117-d]*

> *This lack of confidence in the system was not restricted to the ambulance crews. Control management and staff also had less than complete confidence in the system often because of adverse feedback from ambulance crews due to the factors noted above. There were also many factors concerning the computer system technical reliability and system performance that affected their attitude. [3118]*

It is important to emphasize that the concern towards overall system reliability permeates the report of the inquiry and, in fact, several recommendations were made regarding the future of CAD systems in this respect. Among them are augmenting the overall reliability of the system, improving the level and quality of the provision of ambulance services, and increasing the ownership by all stakeholders (management, staff and ambulance crews). We decided to include those recommendations to illustrate the strong relation this aspect (and we might extend the remark to include all non-functional requirements) to the success of the system.

> *i. it must be fully reliable and resilient with fully rested levels of back-up;*
>
> *ii. it must have total ownership by management and staff;*
>
> *iii. it must be developed and introduced in a timescale, which, whilst recognizing the need for earliest introduction, must allow fully for consultation, quality assurance, testing and training.*
>
> *iv. management and staff must have total, demonstrable, confidence in the reliability of the system.*
>
> *v. the system must contribute to improving the level and quality of the provision of ambulance services in the capital;*
>
> *vi. any new system should be introduced in a stepwise approach, with, where possible, the steps giving maximum benefit being introduced first;" [1009-c]*

5.2 Performance

As defined by Yeh et al [Yeh84] performance describes both subjective and objective qualities of the target system. It is a measure of the success of a system in terms of velocity and at the same time performance can act as a constraint that limits the expected behavior of the system.

In the case of the LAS, it is very clear that the performance of the system played a decisive role in its downfall. The passage bellow gives clear evidence that the software programs were designed for functionality rather than speed, and that their performance was notably slow.

> *"To date they (the programs) have been designed for functionality rather than speed and this area could usefully be revisited." [3128]*

> *"The slow speed of response of the software is also a regularly reported issue." [3126]*

The conclusions of the report on the inquiry confirms that the performance of the system was indeed very poor, stressing the fact that system response times were unacceptable at times. Ironically enough, the report claims that the system "did what it had been design to do:" It is clear from the observation that the developers of the system gave priority to the implementation of the functional requirements (understood as the "what") but little attention to its performance thus failing to perceive that excessively slow response times would end up jeopardizing the system as a whole. This situation is described by the passage below:

> *"On 26 and 27 of October the computer system itself did not fail in a technical sense. Response times did on occasions become unacceptable, but overall the system did what it had been designed to do. However, much of the design had fatal flaws that would, and did, cumulatively lead to all of the symptoms of system failure." [1007-x][4016]*

5.3 Integrity

The system relied on near perfect information on vehicle location and crew/vehicle status. Without accurate knowledge of vehicle location and status the system was prevented from perfect functioning thus becoming unable to allocate optimum resource to an incident. This scenario is illustrated below:

> *"The system relied on near perfect information of vehicle location and crew/vehicle statures. Without accurate knowledge of vehicle location and status the system could not allocate the optimum resource to an incident…it is believed that the majority of allocation errors were due to the system not knowing the correct vehicle location or status of vehicles that may have proved more appropriate." [4008]*

The algorithms for allocating resources did not appear to operate gracefully. On top of that, as the number of accidents increased there was a loss in performance and the system slowed down considerably. It is interesting to notice, however, that none of these facts is pointed out as the actual cause of system failure. The main cause, as the passage bellow illustrates, is the loss of system integrity as a result of lack of information on vehicle location.

> *"… although some poor allocations may be attributable to errors in the allocation routine, it is believed that the majority of allocation errors were due to the system not knowing the correct vehicle location or status of vehicles that may have prove more appropriate." [4008]*

This discussion brings to light a very interesting issue concerning the project of the LAS CAD system, that is why the developers opted for anchoring the system in the input provided by a vehicle location system that is known to be unreliable at times, specially in an urban environment such as London.

> *"There were outstanding problems with data transmission to and from the mobile data terminals" [1001]*

Another interesting aspect of the report on the inquiry on the LAS directly connected to the overall integrity of the delivered system is the underlying assumption that there was no major problems with the technology applied to deliver the system. It was assumed that the system would work properly if the right decisions were taken. This was not the case as there is evidence to complaints regarding some of the basic technologies used during the project, for example, Windows, RIFS (radio interface system), MDT (mobile data terminal), and Datatraks. The passages bellow illustrate this scenario:

> *"Our own simulation of certain Visual Basic routines within Windows 3.0 resulted has resulted in some unexplained system crashes " [3128]*

> *“the system relied on a technical communication infrastructure that was overloaded and unable to cope easily with the demands that CAD would place on it, particularly in a difficult communications environment such as London.”[1007-t]*

> *“(regarding the MTD’s) there has been criticism on the design of these units, particularly their readability. ” [5008]*

> *many problems were identified including inaccurate Datatrak location fixes caused by faulty equipment, transmission black spots, or software error. [3090]*

5.4 Cost

During the procurement stage the management of the LAS seemed to have been primarily concerned with overall cost of the project. The procurement process description reveals several important aspects limited to management, but it brings up interesting aspects regarding price policies in the IT (information technology) industry. A major consulting company states that a package solution would take nineteen months to be implemented at a cost of 1.5 million. The software company that won the bid charged only thirty five thousand for the software and the whole contract came up to 937.5 thousand, 700 thousand cheaper.

As the passages bellow make clear, the emphasis of the standing finical instruction was very much obtaining the best price.

> *(during the procurement process) “The emphasis was very much on obtaining the best price.” [3032]*
>
> *“The standing financial instructions also state that the lowest tender should be accepted unless there are “good and sufficient reason to the contrary”. The lowest tender was taken.”[3031]*
>
> *“the standing instructions provide little qualitative guidance to procurement teams”[3032]*

The precedence of quantitative aspects (best price) over qualitative ones (who would do the job best?) exemplifies what Yeh et all [Yeh 84] describes as the design versus cost to function trade-off. Very few projects are undertaken in which cost is no object; nevertheless some other issues should have been taken in consideration during the procurement such as previous experience in the development of systems of this nature together with the, i.e., ability to perform the tasks required, ability to handle throughput and response times, case of use by staff and resilience. The passage below exemplifies the little weight previous experience had on the cost trade-off:

> *"It is also clear (from the actual procurement) that no specific weighting was given to the extent of supplier experience in Command and Control systems." [3040]*

5.5 Usability/ User Interface

System usability is undoubtedly one of the greatest concerns during system specification and development. A system that presents a badly planned interface is very unlikely to succeed among its users. In recent years the areas of human computer interaction and interface design have largely grown mainly due to the adoption graphical user interfaces (GUI).

The CAD system developed for the London Ambulance Service was no exception. The system was developed in Visual Basic and provided its users with interaction via a GUI. Unfortunately the interface posed a great load on the system, resulting in a significant slow down of more basic operations, as stated below:

> *"the performance of Visual Basic applications is not fast. Filling screens with Visual Basic applications tool takes time measured in several seconds." [3128]*

In this case we notice that two different non-functional requirements of the system, namely usability and performance are in such position that the fulfillment of one results in an obstacle to the fulfillment of the other. The passage below illustrates the situation:

> *"The LAS desire to opt for a graphical user interface as opposed to a text based presentation has resulted in a trade off performance in favor of ease of operator use. [3128]*

During the requirements engineering process, conflicting requirements are bound to arise. To resolve the conflicts, both parties should engage on a negotiation process until a consensus is reached. . Mylopoulos provided for the event of requirement opposition by treating them as goals of the system and modeling those in a framework where it is possible to identify and refine conflicts [Mylopoulos92]. In this light, the author suggests that requirements are not individually fulfilled but a general agreement among them is reached. Nissen et alli present another technique for managing the conflicts using multiple perspectives in [Nissen96] and Barry Boehm in [Boehm96] presents a knowledge based tool that helps clarify and solve such conflicts.

5.6 Communicativeness

Another desirable feature of a software system is its power to promote communication among parties, either human or computer. It is expected that a system should enhance the communication process by providing more accurate and detailed information than its manual counterparts. In the case of the LAS CAD system this expectation was not met, as problems with the automated routine were noticed. The passage below establishes a parallel to the otherwise manual system in this respect

The problem of reporting status, however caused, resulted in wrong allocations being made that were less likely to happen in a voice contact/manual allocation world. [3117]

Bibliographical References

[Antón96] - Antón, A. - Goal-Based Requirements Analysis - in *Proceedings of the Second International Conference on Requirements Engineering,* IEEE Computer Society Press – Colorado Springs, April 14 to 18, 1996, pp.136-144.

[Basili91] – Basili, V., Musa, J. – The Future Engineering of Software: a management perspective – IEEE Computer – Vol.24, No. 9, September 1991, pp. 90-96.

[Berdon95] - Berdon, J. , Davis, A. - Multiple Viewpoint Based Method for Requirements Engineering – paper submitted to the Software Engineering Journal in Dec 1994–University of Colorado at Colorado Springs – 1995.

[Bersoff80] – Bersoff, E. et al. – *Software Configuration Management* – Prentice Hall, 1980.

[Boehm78] – Boehm, B. et alli – Characteristics of Software Quality –North Holland Press - 1978

[Boehm96] - Boehm, B.; In, H. - Identifying quality requirements conflicts - IEEE Software - March 1996, pp.25-36

[Boehm96] - Boehm, B.; In, H. - Identifying quality requirements conflicts - IEEE Software - March 1996, pp. 25-35.

[Bowman96] - Bowman, H., Derrick, J.,,Linington,P. and Steen, M. - Cross Viewpoint consistency in open distributed processsing - *Software Engineering Journal* - January 1996, pp. 44-57.

[Brooks75] - Brooks, F. - *The mythical man-month*, Addison Wesley, 1975

[Davis82] - Davis, G. - Strategies for information requirements determination - *IBM Systems Journal* - vol21 N1 – 1982, pp.4-31.

[DeMarco91] – De Marco, T. – Non-Technological Issues in Software Engineering – Proceedings of the 13th International Conference on Software Engineering (ICSE) – IEEE Computer Society Press – 1991 – pp.149-150.

[Easterbrook94] - Easterbrook, S. -Resolving requirements conflicts with computer supported negotiation - in *Requirements Engineering: Social and Technical Issues* edited by Joseph Goguen and Marina Jirotka - Academic Press 1994, pp.41-66.

[Easterbrook96] - Easterbrook, S. - Nuseibeh, B.- Using ViewPoints for inconsistency management - *Software Engineering Journal* - January 1996, pp. 31-43.

[Finkelstein93] – Finkelstein, A. et. al. - Requirements engineering through viewpoints -

[Finkelstein96] – Finkelstein, A. - A Comedy of Errors: The London Ambulance Case Study - Introduction to the case study for the 8th IWSSD.

[Goguen94] - Goguen, Joseph - Requirements Engineering as the reconciliation of social and technical issues - in *Requirements Engineering: Social and Technical Issues* edited by Joseph Goguen and Marina Jirotka - Academic Press 1994.

[Goguen94] - Goguen, Joseph - *Requirements Engineering as the reconciliation of social and technical issues* - in Requirements Engineering : Social and Technical Issues edited by Joseph Goguen and Marina Jirotka - Academic Press 1994.

[Goguen94b] - Goguen, J.A. and Linde, C.. Techiques for Requirements Elicitation, *In Proceedings of the First IEEE International Symposium on Requirements Engineering*, San Diego, Ca, IEEE Computer Society Press, 1994, pp 152-164.

[Goguen94c] - Goguen, J.A.. Social Issues in Requirements Engineering, *In Proceedings of the First IEEE International Symposium on Requirements Engineering*, San Diego, Ca, IEEE Computer Society Press, 1994, pp 194-195.

[Gotel93] – Gotel, O. and Finkelstein, A. – An analysis of the Requirements Traceability Problem - Imperial College Department of Computing Technical Report TR-93-41 – 1993.

[Gotel95] – Gotel, O. and Finkelstein, A. – Contribution Structures – in the *Proceedings of the Second IEEE International Symposium on Requirements Engineering (RE'95)* – York, March 27 to 29 – IEEE Computer Society Press, 1995, pp. 100-107.

[Gotel97] – Gotel, O. and Finkelstein, A. – Extended Requirements Traceability: Results from an Industrial Case Study– in the *Proceedings of the Third IEEE International Symposium on Requirements Engineering (RE'95)* – Annapolis, January 6-10 – IEEE Computer Society Press, 1997, pp. 169-179.

[IEEE-Std830-1984]– The Institute of Electrical and Electronic Engineers - IEEE Guide to Software Requirements Specifications – ANSI/IEEE Std 830-1984.

[IWSSD96] – *Proceedings of the 8th International Workshop on Software Specification and Design* – IEEE Computer Society Press – Schloss Velen, Germany, March 1996.

[Jacobson94] – Jacobson, I. – *Object Oriented Software Engineering: a use case driven approach* – Addison Wesley, 1994.

[Jirotka95] – Jirotka, M. et al. – Ethnography by Video for Requirements Capture – mini tutorial presented at the in the *Second IEEE International Symposium on Requirements Engineering (RE'95)* – York, March 27 to 29, 1995.

[Kaidl93] – Kaidl, H. – The Missing Link in Requirements Engineering – *Software Engineering Notes* – ACM SIGSOFT - Vol. 18 No. April 1993, pp.30-39.

[Kellner91] – Kellner, M. – Software Process Modeling Support for Management Planning and Control - Proceedings of the First International Conference on Software Process – IEEE Computer Society – October 1991, pp.8-28.

[Kirner96] – Kirner, T. and Davis, A. - Nonfunctional Requirements of Real -Time Systems – *Advances on Computers* – Vol 42, pp.1-37.

[Kling82] - Kling, R., Scacchi, W. - The Web of Computing: Computer technology as Social Organization – Advances in Computing Vol 21. No.1 – 1982.

[Knuth89] – Knuth, D. – The errors of T_eX – Journal of Software Practice and Experience, Vol. 19 No.7, July 1989, pp.607-685.

[Kotonya95] - Gerald Kotonya and Ian Sommerville - Requirements Engineering with Viewpoints - Technical Report CSEG/10/1995, CSEG Computing Department, University of Lancaster, 1995.

[Kramer96] – Kramer, J. and Wolf, A. – Succeedings of the 8th International Workshop on Software Specification and Design – in *Software Engineering Notes* – ACM SIGSOFT Vol.21 No. 5 September 1996, pp. 21-35.

[Lehman80] – Lehman, M. – Programs, Life Cycles and Laws of Software Evolution – Proceedings of the IEEE 68(9), 1980, p. 1060-1076.

[Leite83] – Leite, J.C.S.P. – SADT Datagrams: A Powerful Tool for Requirements Analysis, in *Dept. of Information and Computer Science – University of California, Irvine*, ASE Project RPT 022 (Oct 1983), pp. 1-12.

[Leite87] – Leite, J.C.S.P. – A Survey on Requirements Analysis – Technical Report RTP-071 – *Department of Information and Computer Science, University of California at Irvine,* ASE tech. Report 20/87, (Feb 1987), pp.1-15.

[Leite91] - Leite, J.C.S. P. - Freeman, P. - Requirements Validation Through Viewpoint Resolution - *IEEE Transactions on Software Engineering* Vol. 17 No 12, 1991, pp. 1253-1269.

[Leite95] - Leite, J.C.S.P. and Oliveira, A.P. – A Client Oriented Requirements Baseline - in the *Proceedings of the Second IEEE International Symposium on Requirements Engineering (RE'95)* – York, March 27 to 29 – IEEE Computer Society Press, 1995, pp.108-115.

[Leite96] – Leite, J.C.S.P. – Viewpoints on Viewpoints – in Proceedings of

[Leite97] – Leite, J.C.S.P. et al. – Enhancing a Requirements Baseline with Scenarios - – in the *Proceedings of the Third IEEE International Symposium on Requirements Engineering (RE'95)* – Annapolis, January 6 to 10 – IEEE Computer Society Press, 1997, pp. 44-53.

[Lingaard94] – Lindgaard, Gitte – Usability Testing and System Evaluation – Chapman and Hall – London, 1994

[Luff93] - Luff, P., Jirotka, M., Heath, C. and Greatbatch, D.. Tasks and Social Interaction: the Relevance of Naturalistic Analyses of Conduct for Requirements Engineering, *In Proceedings of the First IEEE International Symposium on Requirements Engineering*, San Diego, Ca, IEEE Computer Society Press, 1994, pp 187-190.

[Luff94] - Luff, P., Jirotka, M., Heath, C. and Greatbatch, D.. Work, interaction and technology: The naturalistic analysis of human conduct and requirements analysis, *In Requirements Engineering: Social and Technical Issues*, M.Jirotoka and J. Gougen, San Diego, Ca,Academic Press, 1994, pp 259-288.

[McCall94] – McCall, J. – Quality factors in Encyclopedia of Software Engineering – vol.2 – J. Marciniak editor, John Wiley & Sons, New York, pp.958-969.

[Mylopoulos92] – Mylopoulos, J. et al. – Representing and Using Non-Functional Requirements: A process oriented approach – *IEEE Transactions on Software Engineering* – Vol. 18 No. 6, June 1992, pp.483-497.

[Nissen96] - Nissen, H.; Jeusfeld, M.; Jarke, M. Zemanek, G.; Huber, H. - Managing multiple requirements perspectives with metamodels - IEEE Software - March, 1996, pp.37-47.

[Nuseibeh94] - Bashar Nuseibeh, Jeff Kramer and Anthony Finkelstein - A framework for expressing the relationships between multiple views in requirements specifications - *IEEE Transactions on Software Engineering*, Vol 20 No.10, October 1994, pp. 760-773.

[Pinheiro96] – Pinheiro, F. and Goguen, J. – An Object Oriented Tool for Tracing Requirements – IEEE Software – Vol. 13, No.2, March 1996, pp.52-64.

[Pinto93] – Pinto, P.E.; Gaio, F. – Análise da adoção de novas tecnologias no processo de desenvolvimento de software da Petrobrás – VII Simpósio Brasileiro de Engenharia de Software (SBES) – Sociedade Brasileira de Computação - Rio de Janeiro, RJ – 1993

[Pohl96] – Pohl, K. – PRO-ART: Enabling Requirements Pre-Traceability – in *Proceedings of the Second International Conference on Requirements Engineering,* IEEE Computer Society Press – Colorado Springs, April 14 to 18, 1996, pp.76-85.

[Potts88] – Potts, C. and Bruns, G. – Recording Reasons for Design Decisions – in Proceedings of the 10th International Conference on Software Engineering – IEEE Computer Society Press – April, 1988, pp.418-426.

[Pressman92] – Pressman, R. – *Software Engineering: a Practitioner's Approach* – McGraw Hill, 1992.

[Randall93] - Randall, D., Hughes, J. and Shapiro, D. Steps towards a partnership: Ethnography and system design, *In Requirements Engineering: Social and Technical Issues*, M.Jirotka and J. Gougen, San Diego, Ca, Academic Press, 1994, pp 241-258.

[RI-LAS93] – Report of the Inquiry into the London Ambulance Service – electronic version prepared by prof. Anthony Finkelstein , available at ftp://ftp.cs.city.ac.uk/pub/requirements/lascase0.9.ps.gz – with permission from the communications directorate, South West Thames Regional Health Authority, Original ISBN: 0 905133 70 6.

[SEI-CMI1990] – Carnegie Mellon University - Software Specifications: A framework- SEI Curriculum Module SEI-CM-11-2.1, January 1990.

[Sommerville93] - Sommerville, I., Rodden, T., Sawyer, P., Bentley, R. and Twidale, M.. Integrating ethnography into the requirements engineering process, *In Proceedings of the First IEEE international Symposium on Requirements Engineering*, San Diego, Ca, IEEE Computer Society Press, 1994, pp 165-173.

[van Lamsweerde95] – van Lamsweerde A. et al. – Goal – Directed Elaboration of Requirements for a Meeting Scheduler: Problems and Lessons Learned, in the *Proceedings of the Second IEEE International Symposium on Requirements Engineering (RE'95)* – York, March 27 to 29 – IEEE Computer Society Press, 1995, pp.195-203.

[Walz93] – Walz, D., Elam, J. Curtis, B. – Inside a software design team: Knowledge acquisistion, Sharing and Integration – CACM, Vol. 36, No.10, 1993, pp.62-77.

[Wood94] - Wood, D.P., Christel, M.G. and Stevens, S.M., A Multimedia Approach to Requirements Capture and Modeling, *In Proceedings of the First International Conference on Requirements Engineering* IEEE Computer Society Press – Colorado Springs, April 18 to 22, 1994, pp.53-58.

[Yeh84] - Yeh, R.; Zave, P.;Conn, A.; Cole, G. - Software Requirements: New Directions and Perspectives - *Handbook of software engineering* – 1984, pp.519-543.

[Zave97] – Zave, P., Jackson, M. – Four dark corners of requirements engineering – ACM Transactions on Software Engineering and Methodology Vol.6 No.1, January 1997, pp. 1-30.