ISSN 0254-2757

# QUÆSTIONES INFORMATICÆ

Volume 6 • Number 3 November 1988



The official journal of the Computer Society of South Africa and of the South African Institute of Computer Scientists

Die amptelike vaktydskrif van die Rekenaarvereniging van Suid-Afrika en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes

# QUÆSTIONES INFORMATICÆ

**The official journal of the Computer Society of South Africa and of the South African Institute of Computer Scientists**

**Die amptelike vaktydskrif van die Rekenaarvereniging van Suid-Afrika en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes**



## Production

Mr Q H Gee
Department of Computer Science
University of the Witwatersrand
Johannesburg
Wits
2050

## Subscriptions

The annual subscription is

| | SA | US | UK |
|---|---|---|---|
| Individuals | R20 | $7 | £5 |
| Institutions | R30 | $14 | £10 |

to be sent to:
*Computer Society of South Africa*
*Box 1714 Halfway House 1685*

**Quæstiones Informaticæ is prepared by the Computer Science Department of the University of the Witwatersrand and printed by Printed Matter, for the Computer Society of South Africa and the South African Institute of Computer Scientists.**

# Learning the First Step in Requirements Specification

R M C Short
*Department of Computer Science, University of the Witwatersrand, P O, Wits, 2050*

## Abstract

*Learning about the domain into which an information system will fit is an essential part of the system development process.*
*In this paper it is argued that the first stage of the development process should be learning undertaken by the designers in order to come to an understanding of the domain into which the information system is intended to fit. More specifically it is argued that this understanding must be an understanding of the "intended information system's user's" own view of the domain. Based on this contention a suggested sequence of foci to be adopted during the specification of information requirements is proposed.*

Received August 1988, Accepted October 1988

## 1. Introduction

Every human artefact is a projection of thought. Confusion and muddle will naturally project a poor quality artefact. An information system is a human artefact. It is important to get the thinking on which it is to be founded correct. Requirements specification is the phase of development which is dedicated to this task.

Requirements specification is generally recognised as an absolutely crucial stage in the development of any information system and yet there are as yet no known infallible ways of establishing the requirements.

The very nature of the situation is probably such that there never can be an absolutely infallible way of determining requirements. Infallible that is in the sense of producing a correct and complete set of requirements. Not that a finally complete set of requirements can in any event ever be achieved. The reason for this is the volatile nature of the business environment subject, as it is, to continuous change and this in turn leads to changes in the information requirements. There are however less successful and more successful approaches to the specification of requirements at a point in time. Davis [4] has presented a strategy for selecting the approach best suited to a particular situation.

This paper is founded on the belief that there are less effective and more effective approaches to requirements specification. It is argued herein that the conscious and deliberate execution of an activity, implicit in the requirements specification of information systems which give user satisfaction, will help to improve performance generally in requirements specification.

In pursuit of this objective, requirements specification is first put into context in the system life cycle. Then some concepts which are central to the following discussion are defined, user satisfaction and requirements *per se* are explored before considering the usual foci in requirements specification and suggesting some slight changes in the usual application of these foci.

## 2. Position in System Life-Cycle of Requirements Specification

For more than fifteen years it has been generally accepted that the development from inception through to operational use of an information system involves the execution of a number of distinguishable phases in a known sequence. This is known as the system life cycle.

The individually recognised phases of the life cycle can be grouped together into larger stages termed respectively Specification, Development, Implementation, and Operation & Maintenance [5].

Specification embraces the phases of Problem Definition, Feasibility Study, and Requirements Specification.

Development embraces the phases of System Design, Program Design and Code, Program and System Test, and Manual Procedures Development.

Implementation embraces User Training, user Acceptance Testing, and Production Installation.

Operation and Maintenance embraces the operational use of the system as well as its continuous adjustment and enhancement to suit changing user requirements.

This paper addresses one aspect of Requirements Specification which is one of the phases within the Specification stage.

Requirements Specification, even though it is preceded by the Problem Definition and Feasibility

Study phases within the Specification stage is nevertheless the foundation phase in the development portion of a systems life cycle. In fact development proper begins with Requirements Specification.

We turn now to an exploration of what is involved in Requirements Specification and an exploration of how the quality of its end result might be made more certain than at present. But first it is necessary to clearly define some terms that are central to this exploration.

## 3. Information Systems, Object Systems, Users and Developers

For the purpose of this paper an **information system** is defined as a data processing system which is designed to output data that yields information about the states of some other system.

This other system, after Davis [4], we refer to as the **object system.**

The information system yields information about the object system states by means of modelling the object system [10].

The information system is used by actors on and within the object system to gain information about states of the object system in order to direct and control it in various ways. In this paper these actors are referred to as **users** or **the users.**

The system created through the interaction of the information system, the users and the object system is referred to as the **task domain** in this paper.

Those responsible for carrying out the specification of the requirements of the information system are termed the developers or developer. The developers' role is regarded as quite distinct and separate from the users' role.

## 4. Requirements Specification and User Satisfaction

The specification of the requirements for a proposed information system is neither a simple nor trivial task.

It is not a simple task because of the great multiplicity of interrelated issues and factors that have to be considered. It is also not a trivial task because its end result is the set of design objectives to which the developed information system is intended to conform.

It is absolutely essential that the satisfaction of these design objectives should in its turn result in an information system which satisfies its intended users. To the degree that the users are not satisfied with the product even though it meets the design objectives then to that degree the development effort can be said to have been misdirected and therefore wasted. The misdirection of the effort in such cases can only have arisen from inadequate or inappropriate design objectives and they in their turn are the result of failures in requirements specification.

McKeen [12] in a field study of the effort expended on the development of 32 business application systems found that the level of user satisfaction varied in direct proportion to the amount of effort expended in the Specification stage as opposed to the Development stage in the production of the application systems.

At the present time ways and means of ensuring zero failures in requirements specification are not known. There are however a number of different ways of determining requirements and Davis [4] has suggested a means by which the most effective way for a particular situation can be selected.

In this paper it is suggested that no matter which of the approaches to Requirements Specification is adopted it will have an enhanced chance of achieving ultimate user satisfaction if it is prefixed by an explicit attempt by the developers to come to an understanding of how the users understand the task domain of which the information system will form a component.

Greenspan [8] recognises the importance of domain knowledge in software development contending that an understanding of this issue will lead to greatly improved software development practices. He does not however explicitly state whose domain knowledge he is referring to; that of the users presumably, but it could also be to some implicitly assumed abstract absolute. In the author's experience this latter unspoken assumption is often operational in practice – he once held it himself.

Langefors [11] in discussing information's relationship with data says "data alone cannot carry information, they can only, at best, give rise to information in the minds of people and only in those people who hold a suitable frame of reference, or world view, or receiving structure in the mind."

Giddings [7] says "The complexities associated with designing an easy to use system are related to human factors and cognitive psychology more than to data processing."

The obvious implication of the foregoing is that the developers need to come to an understanding of the users' conceptualisation of the task domain in addition to and distinct from their own understanding if they are to be able to develop a system which gives user satisfaction.

## 5. Requirements, what are they?

There is, in the view of the author, a fundamental lack of conceptual clarity amongst many within the information systems discipline as to exactly what is being referred to when we talk of requirements. This is undoubtedly because requirements embrace a very wide spectrum of issues.

In a paper titled "A Taxonomy of Current Issues in Requirements Engineering" Roman [15] identifies two major groupings of requirements.

The first category he labels functional and the second non-functional. It is interesting to note the use of the term functional here. It is an indicator of the information industry's historical and initially necessary pre-occupation with data processing functions rather with than their product namely information. In contrast the chief pre-occupation of this paper is with information requirements *per se* and their specification.

In respect of the specification of the functional requirements Roman says that "it involves the modelling of the relevant internal states of both the component and its environment."

This statement and the further narrative linked to it encompass an understanding of the nature of requirements which is worthy of further consideration in order to draw out some useful finer distinctions.

First, for the particular purpose of this paper the word **component** in Roman's description of functional requirements specification is taken to refer to a whole information system and the word **environment** is taken to refer to the object system as well as the user's conceptualisations of the object system.

In respect of the modelling referred to in the description, Roman goes on to quote Balzer and Goldman [2] who have noted that the model (of requirements) must be cognitive in nature and embrace concepts that are relevant to the milieu in which the component is used. The concepts must not be related to the design or implementation of the component they say.

Balzer and Goldman's use of the word **milieu** here rather than the word **environment** may or may not signify anything. However it could have been used because Balzer and Goldman wanted to refer to more in this phrase than the object system pure and simple. They wanted perhaps to also capture or include the users' and their conceptualisations of the object system, This latter interpretation is the one which appeals especially as they say that the model of requirements must be cognitive in nature.

If this interpretation of the meaning of the word milieu, as used by Balzer and Goldman, is correct then what they are saying supports the view that the users' conceptual model of the object system is of significant importance in the specification of requirements.

In fact it is probably true to say that to the degree that the concepts reflected in the requirements are not those of the users then to that degree the users will not feel comfortable with the resultant information system and may not even be able to understand the information that it produces.

Roman puts it this way, he says that if the environment (milieu) is not well understood (by the developers) it is unlikely that the requirements (as specified by them) will reflect the actual needs the component must fulfil.

Ahituv [1] when discussing the value of information says "value cannot be assigned to a set of data, but only to the set of concepts perceived when the data are displayed to the decision-maker."

In other words the data supplied by the information system must relate to the concepts that the users hold in respect of the object system. If they do not the users will be unable to glean meaningful information about the state of the object system from them.

Shemer [16] also emphasises the need to understand the object system. He says "requirements specification should reflect an understanding of a system (object system), guide the subsequent design and programming phases and serve as a basis for all communications concerning the software system (information system) being developed."

He does not however draw a distinction between the users' understanding and the developers' understanding as such, assuming, presumably, that these will be the same. Such is, in the nature of things, not necessarily the case however.

Borgida, Greenspan and Mylopoulos [3] also see requirements specification as modelling. They say "the task of requirements specification thus has at its core the building of a requirements model for some portion of the world." They then go on to say "that most information systems being developed reflect the designer's perceptions of the world." They do not however go on to point out the obvious problem inherent in this which is that the users deal with this world in terms of their own perceptions not necessarily those of the designers.

Returning once again to Roman. The purpose of modelling the internal states of the environment (object system only in this case) is to enable the information system, built according to the requirements, to give to the user relevant information about the states of the environment. In his description of functional requirements Roman does not make so direct a statement as the foregoing but that it is as stated therein is surely unquestionable.

Thus it can be argued that Roman's functional requirement specifications are first and foremost statements about the users' information needs in respect of the object system. They are only secondarily about any information system processes deemed to be necessary to provide the information. This fact is worth emphasising because different processes can yield equivalent information. A number added to itself produces the same result as multiplying it by 2 for instance.

Thus Roman's label **functional requirements** is a misnomer for this grouping of requirements. The grouping should really be divided into two, information requirements and functional requirements. From henceforth when referring in this paper to functional requirements as defined by

Roman they will continue to be referred to as **Roman's functional requirements** whilst the term **functional requirements** will be used to refer to processing requirements as distinct from information requirements. The specification of information requirements takes natural procedural precedence over the specification of functional requirements. This is because it is the users' need to know about the states of the object system which generates the information requirements and the functional requirements then arise naturally from the need to produce the specified information.

As stated earlier it is essential that the specified information requirements match with the users' conceptualisations of the object system. Specified functional requirements that refer to functions of which the user needs to be aware must match with users' conceptualisation of the task domain. Other specified functional requirements have no need to match with users' concepts but they must in no way damage the integrity of concept match for those that do need to match.

Roman's second grouping of requirements is labelled non-functional. By non-functional requirements is meant general real world constraints on the design such things as performance, costs etc. fall within non-functional requirements. These requirements are not considered in this paper. It is probable however that the approach to requirements specification that is proposed here could make a contribution to their specification.

Non-functional requirements are secondary to Roman's functional requirements but, they are never-the-less important. They might mean that an information system cannot be built because of an inability to satisfy some of them e.g. cost constraints.

Having considered what is meant by requirements we turn now to a consideration of the process involved in specifying Roman's functional requirements.

## 6. Requirements Specification – the Foci

The process of requirements specification has traditionally tended to have a single or main focus. This focus has altered with the evolution of the information systems discipline.

At the beginning of the evolutionary process the focus tended to be on "the how" of producing information, the process in other words. Hence the term functional requirements already discussed.

The focus then shifted more towards "the what", what is the information that is to be produced. This particular focus was fuelled by the development of database technology which gave a large increase in the ease and facility with which data could be manipulated to produce information.

In more recent times there has been an increasing recognition that "the why" needs to be addressed. Namely, why is the information needed in the first place, for what is it wanted. This is exemplified by the growing recognition of information as a strategic weapon in the market place [9].

In reality all three foci are important and should be adopted by the developers in the reverse sequence to that of their historical recognition as given above.

Thus the first focus that should be addressed is "why is the information needed" and the second should be "what information is needed" and the third should be "how is the information to be produced".

Suitable conditions for addressing the first focus, "the why" are the concern of the next section.

## 7. Conditions for Addressing *The Why* Focus

In order to be able to adopt "the why" focus when specifying information requirements this paper contends that developers need to develop a number of distinct understandings. And in order for these understandings to be able to develop it further contends that the developers need to accept and acknowledge the existence of the following distinctions:

a) the object system
b) the current information system
c) their own understanding of a)
d) their own understanding of b)
e) the users' understanding of a)
f) the users' understanding of b)

If the developers are not consciously aware of these distinctions many conceptual confusions can arise in the developers as well as mismatches between the developers' concepts and those of the users [16].

The particular set of mismatches that the approach suggested in this paper is intended to prevent are where the developers recognise the distinct existence of a) and b) but do not distinguish c) to f).

The consequence of recognising only a) and b) is that the developers then operate on the unspoken assumption that how they see (conceptualise) a) and b) is how they actually are and therefore how the users see them.

When this is actually the case i.e. there are no mismatches between the users' and developers' concepts then it is likely that the resultant information system will be acceptable to the users as it will fit with their conceptual models of the object system and the task domain.

However if this is not the case then to the degree that it is not so then the resulting information system will be found to be unsatisfactory by the users due to conceptual mismatches between their understandings of the object system and task domain

and those reflected in the information system.

## 8. The Current Chief Focus and its Problems

The current predominant focus in information requirements specification is on "the what". This focus should ideally come after "the why" focus. It should not take up the whole field of view and exclude it. Recognition of distinctions a) to f) is not a strong or really necessary characteristic of "the what" focus. Consequently it has potential for leaving hidden mismatches between the users' conceptual model of the object system and that of the developers.

Enmeshed in this focus developers have sought to determine from the users what information they will require, when they will require it and how they will require it to be presented.

Now there is nothing intrinsically inappropriate about determining the above things. It is in fact essential that they should be made explicit at some point there are however two unfortunate issues which arise as a result of this single minded focus.

Firstly it evades the fact that these things are consequences not causes. They are the consequences of the users' need for information about the object system in order to better direct and control it. In view of this fact it is surely better that the users' understanding of the object system should be the first focus of study. Otherwise an appropriate model of it, in the shape of an information system, is not likely to be built. It is putting the cart before the horse to focus study on the information to be output by the model rather than to begin with a study of what is to be modelled.

In addition going straight for the users' perception of their information requirements skips past the opportunity to perhaps uncover any weaknesses, inconsistencies and gaps in the users' conceptual models of the object system and to remedy these by suggesting improvements.

Thus the current single-minded focus on the output information has unfortunate potential consequences. Luckily however as anybody who has been involved in requirements specification whether with an information or a functional focus will know, in the process of determining the requirements it is inevitably necessary to learn about the object system in order to be able to make logical sense of the requirements uncovered.

This learning however does not entail recognising the distinction between c) and e) and between d) and f), i.e. between the developers' and the users' conceptual models thus it does not necessarily result in specifications which yield an information system that matches the users conceptual model of the object system or task domain.

## 9. Some Foci in the Literature

There are few situations where a predominant "what" focus has not been wholly the case. One of these is where a number of information systems are to utilise a common data base [17]. Here there is active recognition that the purpose of the data base is to represent some aspects of the world, not the other way round, and thus this world must be understood before an effective data base can be constructed. Nevertheless even in this situation the first objective right from the start is to define suitable data rather than to understand the object system *per se* let alone the user's conceptual models of it and the task domain.

Davis [4] sees the basis of the users' information needs as lying either in the nature and characteristics of

the system to be served
or the users of the information system
or the interplay of these two factors.

Davis has described various different ways of determining information requirements. He also presents a strategy for selecting the most appropriate way to follow in a particular situation. This strategy for selection takes cognisance of both user characteristics and the characteristics of the system to be served.

Mumford [14] says that both organisational and user personal factors are central factors in the formation of information requirements but she does not specifically make reference to the users' conceptual models of the object system and task domain.

We turn now to suggest an approach to requirements specification which should ensure that the specified information requirements are more likely to correspond with the users' conceptual model of the object system and thus lead to greater user satisfaction.

## 10. A Suggested Sequence of Foci for Requirements Specification

Virtually all authors consistently affirm that user involvement in requirements specification is a key contributor to the success of the process. De Brabander [6] cites eight authors who support this view.

Various reasons are advanced as to why this should be so. Human information processing limitations (Davis [4]), job satisfaction and motivation (Mumford [14]) match between information system and users' view of how its environment operates (Montazemi and Conrath [13]).

This paper shares the view of Montazemi and

Conrath arguing that the users' conceptual model of the object system is a crucial determinant of the information requirements. It is also argued that the reality of this model is one of the chief reasons why the involvement of the user in the specification of requirements is found to be so important and necessary.

Thus it is suggested that the performance of developers in requirements specification would be of more certain quality if developers recognised the distinct existence of a) to f) and conducted requirements specification with the following sequence of deliberate shifts in focus:

1. Activity focuses on developing, an understanding of
   - the object system
   - the current information system
   - the users' conceptual model of the object system
   - the users' conceptual model of the current information system

   This focus enables the developers to enter steps 2 to 4 mentally well prepared to carry them out.
2. Activity focuses on the specification of "the why" and "the what" of the information required by the users always maintaining a good match with the user's conceptual model of the object system (which should be adjusted as proves necessary).
3. Activity focuses on the specification of the necessary logical functions, "the how"; always, for those functions with which the user interacts directly, maintaining a good match with the users conceptual model of the task domain (which should be adjusted as proves necessary).
4. Activity focuses on the specification of the non-functional requirements.

A suggested technique to assist with the execution of step 1 above is Semantic Nets, which will be addressed in a future paper.

## References


[1] N Ahituv, [1987], A Metamodel of Information Flow: A Tool to support (Information Systems Theory, *Communications of the ACM*, **30** (9), 781-791.

[2] R Balzer and N Goldman, [1979], Principles of Good Software Specification and Their Implications for specification Languages, Proc. spec. Reliable Software Conference, Apr. 1979, 58-67.

[3] A Borgida, S Greenspan, and J Mylopoulos, [1985], Knowledge Representation as the Basis for Requirements Specifications, *Computer*, **18** (4), 82-90.

[4] G B Davis, [1982], Strategies for Information Requirements Determination, *IBM Systems Journal*, **21** (1), 4-31.

[5] G B Davis, and M H Olson, [1984], *Management Information Systems: Conceptual Foundations, Structure and Development*, McGraw-Hill Book Co., Singapore.

[6] B De Brabander and A Edstrom, [1977], Successful Information System Development Projects, *Management Science*, **24** (2), 191-199.

[7] R V Giddings, [1984], Accommodating Uncertainty in Software Design, *Communications of the ACM*, **27** (5), 428-434.

[8] S J Greenspan, [1986], On the Role of Domain Knowledge in Knowledge-Based Approaches to Software Development, *ACM SIGSOFT Software Engineering Notes*, **11** (4), 34-36.

[9] B Ives, and G P Learmonth, [1984], the Information System as a Competitive Weapon, *Communications of the ACM*, **27** (12), 1193-1201.

[10] M A Jackson, [1983], *System Development*, Prentice-Hall International, Englewood Cliffs, New Jersey.

[11] B Langefors, [1980], Infological Models and Information User Views, *Information Systems*, **5** (1), 17-32.

[12] J D McKeen, [1983], Successful Development Strategies for Business Application Systems, *MIS Quarterly*, **7** (3), 47-65.

[13] A R Montazemi, and D W Conrath, [1986], The Use of Cognitive Mapping for Information Requirements Analysis, *MIS Quarterly*, **10** (1), 45-55.

[14] E Mumford, [1985], Defining System Requirements to meet Business Needs: A Case Study Example, *The Computer Journal*, **28** (2), 97-104.

[15] G-C Roman, [1985], A Taxonomy of Current Issues in Requirements Engineering, *Computer*, **18** (4), 14-22.

[16] I Shemer, [1987], Systems Analysis: A Systematic Analysis of a Conceptual Model, *Communications of the ACM*, **30** (6), 506-512.

[17] A T Wood-Harper and G Fitzgerald, [1982], A Taxonomy of Current Approaches to Systems Analysis, *The Computer Journal*, **25** (1), 12-16.



This paper was presented at the SACLA 88 conference.

# NOTES FOR CONTRIBUTORS

The purpose of the journal will be to publish original papers in any field of computing. Papers submitted may be research articles, review articles and exploratory articles of general interest to readers of the journal. The preferred languages of the journal will be the congress languages of IFIP although papers in other languages will not be precluded.

Manuscripts should be submitted in triplicate to:

Professor ~~J M Bishop~~ D.C. Kaire
Department of Computer Science
University of the Witwatersrand
Johannesburg
Wits
2050

## Form of manuscript

Manuscripts should be in double-space typing on one side only of sheets of A4 size with wide margins.

The first page should include the article title (which should be brief), the author's name and affiliation and address. Each paper must be accompanied by an abstract less than 200 words which will be printed at the beginning of the paper, together with an appropriate key word list and a list of relevant Computing Review Categories.

Manuscripts may be provided on disc ~~using any Apple Macintosh package or~~ in ASCII format, once that a submitted paper has been accepted.

For authors wishing to provide camera-ready copy, a page specification is freely available on request from the Editor.

## Tables and figures

Tables and figures should not be included in the text, although tables and figures should be referred to in the printed text. Tables should be typed on separate sheets and should be numbered consecutively and titled.

Figures should also be supplied on separate sheets, and each should be clearly identified on the back in pencil with the authors name and figure number. Original line drawings (not photocopies) should be submitted and should include all the relevant details. Photographs as illustrations should be avoided if possible. If this cannot be avoided, glossy bromide prints are required.

Charges

## Symbols

Mathematical and other symbols may be either handwritten or typewritten. Greek letters and unusual symbols should be identified in the margin. Distinction should be made between capital and lower case letters; between the letter O and zero; between the letter I, the number one and prime; between K and kappa.

## References

References should be listed at the end of the manuscript in alphabetic order of the author's name, and cited in the text in square brackets. Journal references should be arranged thus:


[1] E. Ashcroft and Z. Manna, [1972], The Translation of 'GOTO' Programs to 'WHILE' programs, *Proceedings of IFIP Congress 71,* North-Holland, Amsterdam, 250-255.

[2] C. Bohm and G. Jacopini, [1966], Flow Diagrams, Turing Machines and Languages with only Two Formation Rules, *Comm. ACM,* **9**, 366-371.

[3] S. Ginsburg, [1966], *Mathematical Theory of Context-free Languages,* McGraw Hill, New York.


## Proofs

Proofs will be sent to the author to ensure that the papers have been correctly typeset and *not* for the addition of new material or major amendment to the texts. Excessive alterations may be disallowed. Corrected proofs must be returned to the production manager within three days to minimise the risk of the author's contribution having to be held over to a later issue.

Only original papers will be accepted, and copyright in published papers will be vested in the publisher.

## Letters

A section of "Letters to the Editor" (each limited to about 500 words) will provide a forum for discussion ~~of recent problems~~.