The role of requirements analysis in the system life cycle

by YUZO YAMAMOTO, RICHARD V. MORRIS, CHRISTOPHER HARTSOUGH, and E. DAVID CALLENDER
California Institute of Technology
Pasadena, California

ABSTRACT

One of the problems that personnel from the computer industry face today is to find the proper role of requirements analysis in the design and implementation of information-intensive systems so that the results of that activity may be effectively transferred to the rest of the life cycle. This paper addresses the problem by examining the life cycle process in terms of the various viewpoints that human beings use. The interplay between human capabilities and limitations for dealing with the problems of design representation and the increasing complexity of modern information-intensive systems is discussed. The concept of viewpoints around a life cycle wheel that are used throughout the entire life of the information-intensive system is introduced and used to define the functions performed during requirements analysis. Finally, the concept of a system-engineered set of techniques and tools to support the life cycle activities is proposed.
INTRODUCTION

The scope of this paper is the design and implementation of information-intensive systems. Such a system is one where the use and production of information is either a major function or a major component of the control of the process. Such a system usually has as its components hardware and human beings, using software and procedures, respectively.

Historical Background

The development of information-intensive systems is still a difficult task despite progress in relevant methods and tools. One of the early developments in attempts to better manage the development process was the recognition of a system life cycle. This allowed a phased approach in managing the development of information-intensive systems from the definition of initial requirements through operation.

Historically, the phases receiving the most attention have been implementation, integration, and test. The importance of the early phases has been recognized partly because of the increased attention to the entire system life cycle (as opposed to preoccupation with a particular part of it). These early phases are called by various names, such as requirements analysis, logical design, or systems analysis. A few tools are being used during the early phases. For example, successful applications of PSL/PSA have been reported.

One problem encountered by users of requirements analysis techniques is to determine how to transfer the results of the requirements analysis to the later phases of the life cycle. In order to accomplish this successfully, the execution of the system life cycle itself must also be viewed as an information-intensive system and the role of requirements analysis identified in that respect.

Objectives of This Paper

The main objective of this paper is to identify the proper role of requirements analysis in the system life cycle. The second objective is to present an improved life cycle model based on the concept of viewpoints. The third objective is to present the concept of a “techni-kit” to support the life cycle activities.

Structure of the Paper

First, the fundamental causes of the problem, i.e., the complexity of modern information-intensive systems, and the limitations of human beings in handling a complex situation directly are discussed. Next, the viewpoint dimension of characterizing the distinct activities of the life cycle is introduced. Based on the viewpoint dimension, the life cycle wheel model is introduced, and the role of requirements analysis is defined in that framework. Finally, a life cycle support facility, called a techni-kit, is proposed.

HUMAN CAPABILITIES AND LIMITATIONS

We believe that a major source of problems faced during the design and implementation of an information-intensive system is the conflict between the complexity of such systems and human limitations in concurrently handling large amounts of complex information.

Today's information-intensive systems are complex. Instances of complexity can be seen in a number of ways. Typically, systems are made up of many parts, related to one another in many ways. The same is true of many of their components. These systems are often required to meet concurrently a number of distinct but interacting needs. Moreover, a system can often be operated in any of several different ways to meet a particular need.

On the other hand, there are characteristic limitations to human capability in dealing simultaneously with large numbers of concepts and relations. Over a period of time a person can deal with a great number of intricately related concepts, but with only a few at any particular time. There is considerable evidence that indicates that this limit is a small number. This human limitation directly affects the ways in which complex system developments and applications can be conducted.

Methods and tools that are developed to support designers and implementers of information-intensive systems must take explicitly into account two conflicting properties: complexity and human intellectual characteristics. One way to do this is by using (1) conceptual models that provide structure and (2) computer-based tools that augment human logical and clerical capabilities. In this manner the designer can focus individually on a series of small parts of the system with the expectation that the pieces of the system can be combined into an integrated whole. To aid the designers and implementers we have extended the life cycle model and provided a structure for the methods and tools they use.

DIMENSIONS FOR DESCRIBING DEVELOPMENT PROCESS

Any model of the life cycle process must include the following distinct dimensions:

1. Phases: When in time an activity occurs.
2. System level: Where in the hierarchy of the system structure attention is being focused.
As Figure 1 indicates, when a physical object is being examined, the observer's particular location relative to the object is one of the factors that determines the subset of information that the observer can see. The location of the observer is thus a part of the filtering criteria.

Important aspects of the process of taking a view include:

1. The location of the observer (vantage point)
2. The other filtering criteria (process)
3. The data to be filtered (input)
4. The resultant view (output from the viewing process)

The concept of viewpoint will be used as a basis for a model of system development process in the next section.

**LIFE CYCLE WHEEL MODEL OF SYSTEM DEVELOPMENT**

In conventional life cycle models, the role and context of the individual has been largely ignored. To correct this omission, a structure of viewpoints is associated with the development process. The six top-level viewpoints, which form the life cycle wheel model of system development, are illustrated in Figure 2 and defined below. Such viewpoints can most easily be introduced in terms of the activities associated with them. It is important, however, to note that the essence of the viewpoints is not the activities but the continued, vested interest that a group of people develop as they undertake their normal activities. To facilitate understanding, an analogy of the process of building a manufacturing facility for a particular product is given for each viewpoint.

The user needs viewpoint is taken when unanalyzed user needs are sought, captured, and recorded. This may also be called the buyer's viewpoint. This viewpoint is interested in results—solutions to the specific problem presented. The viewpoint also sees and levies constraints under which the system must operate. Individual users often express their needs in terms of scenarios. A scenario describes how the system will operate to achieve a particular purpose and the situation and environment in which the system will be applied.

It is also a temporal exposition of the interdependent activities...
of the environment, the system, and its operators in the accomplishment of a particular purpose. In addition to scenarios, individual users may have needs or desires for certain other characteristics of the system, such as throughput capability, mean time between failures, and adaptability. In a manufacturing environment, the analogous activity is market research.

The user design viewpoint is principally concerned with extracting a common, logically consistent set of requirements from those expressed by individual users. From this viewpoint, system requirements are defined. These requirements provide an envelope of services that encompass all reasonable/acceptable demands by individual users. From this viewpoint, assessments are also made of the stability of requirements and of the effect that changes in requirements will have on the system design. The manufacturing analogy for this viewpoint is product design.

The implementation design viewpoint is principally concerned with laying out an overall structure, or architecture, of components to meet user design requirements. Sometimes work done from the implementation design viewpoint directs the attention of the user design viewpoint to an incomplete area in the design. This work leads to derived requirements that must be incorporated into the user design. Although frequently perceived as different, derived requirements differ from other user requirements only in their date of discovery. The same type of analysis needs to be performed on all user requirements. The manufacturing analogy is the engineering of the manufacturing plant.

The detailed design, implementation, and training viewpoint is concerned with the specifics of constructing the system according to the architecture developed from the implementation design viewpoint. Training is included here because it provides the human components of the system with the guidelines and procedures they will use during system operation. The manufacturing analogy is the construction of the plant and training of the workers.

The integration and test viewpoint is concerned with the fitting together of implemented (or trained) parts of the system to produce a correctly-operating, verified, and validated whole that is ready for use in achieving the users' objectives. In manufacturing, pilot production and product test are the closest analogies.

The user operations viewpoint is concerned with applying the system, as constructed, to achieve the users' objectives. The experience gained in this viewpoint is a rich source of user needs for the next (or modified) system. Use of the plant to produce the product in volume is the manufacturing analogy.

One common viewpoint has been omitted from the present picture of the model—the maintenance viewpoint. The model includes maintenance as a part of operations and restricts its scope to

1. Identifying and evaluating problems within the system
2. Adjusting previously designed controls to keep the system in tune
3. Replacing failed parts with identical spares

If changes in the design of the system or its component are needed, the design is recycled as necessary through the life cycle to design and implement the changes.

There is another important viewpoint associated with the development process, the viewpoint of project management. This viewpoint is illustrated in Figure 3. Project management is able to plan and control the development process by viewing and monitoring the efforts of the major inline development activities performed from the six viewpoints.

The viewpoints have been described in terms of the activities performed. However, as pointed out earlier, the essence of the viewpoints is not the activity, but the continued, vested interest of a group of people involved in the life cycle activities. Two important concepts that arise out of the life cycle wheel model are identified and explained in the following subsections: information transfer and locus of principal activity.

Information Transfer Between Viewpoints

The first concept derived from the life cycle wheel model is that of information transfer between viewpoints. Since the context in which an individual works has been formalized, it is necessary to consider how information is transferred between different viewpoints. When transfers of information are made between viewpoints that are not adjacent, our experience indicates that misinterpretations and misuses of the information are more likely. For this reason, only information transfer between two adjacent viewpoints is considered.

Information exchange and information mappings are the two important kinds of information transfer. An information exchange is a formal process of handing information from one viewpoint to another. It can be further subdivided into feedforward and feedback. Feed-forward is the transformation of a particular element of information from a viewpoint to the next viewpoint (along the general direction of the locus of principal activity, described in the next subsection), and feedback is the reverse. Mappings are relationships established
between the contents of information items of two viewpoints. A consistency check or requirement traceability between views at different viewpoints is an important use of such mappings.

Locus of Principal Activity

The second concept is that of locus of principal activity. The locus of principal activity is the time-ordered sequence of major development activities. It is the replacement for the idealized concept of life cycle phases used in the conventional life cycle model. If the development process were to proceed in the idealized manner, then the development phases would correspond directly to the six major viewpoints. Rarely does the idealized occur. Iteration and concurrent activities make the idealized model invalid.

The separation of viewpoints from life cycle phases is one of the major differences between the life cycle wheel model and the classical development model. It has important implications. First, the viewpoint structure exists throughout the complete life cycle of the information-intensive system. Second, the separation allows the model to explain and show the place for iteration between development activities and concurrent development activities.

The locus of principal activity is the bridge between viewpoints and development phases. Treating phases in this manner focuses project management’s attention on several factors, all of which require management:

1. The existence of iteration in the development process
2. The oscillation of the focus of activity back and forth between viewpoints
3. The existence of concurrent development activities (at least two major efforts existing simultaneously in time)
4. Ensuring consistency between the work done from different viewpoints (requirements traceability)

REQUIREMENTS ANALYSIS IN THE LIFE CYCLE WHEEL

Based on the life cycle wheel model, requirements analysis is viewed as a design activity from a user viewpoint. This design is synthesized from various (incomplete, inconsistent) user scenarios and other expressions of needs. The emphasis is on what functions the system is to perform and how the system interacts with the users. It is helpful in distinguishing between user design and implementation design to say that, from a user design viewpoint, the functions are performed “as if by magic.” A major pitfall to be avoided is the temptation to specify implementation details in the statement of the user design. Use of implementation information should be allowed only as a communication aid or for enumeration of constraints that the users impose. An example of such a constraint would be a de facto selection of computer hardware. This type of implementation design information should be considered as implementation information that appears during the user design phase. To further illustrate the importance of being able to distinguish between views and phases, we point out that it is impossible to finish the user design (in the form of user how-to manuals) until the final stages of implementation.

Requirements analysis (user design) is an important activity at both the information systems level and the software engineering level. As an increasing number of system levels are added to the description of the product system, requirements analysis will first (going along the locus of principal activity) be performed at the information system engineering level for the top system levels of the product system. Later, requirements analysis will again be performed at software subsystem levels. The user design at the system level is a major part of the user needs at the software system level. A classic problem of requirements analysis is ensuring that an internally consistent and complete user design has been prepared. The model delineated in the previous section provides two major features to assist with this problem. The first is the application of consistency mappings to verify that all the scenarios can be supported by the functional features of the user design. In this manner the user design may be tested for completeness using the user scenarios. Internal consistency is a more difficult test to apply. The application of the life cycle wheel model will assist the designer in the check for internal consistency. However, additional design rules that are dependent on the product system need to be applied. The ultimate test for internal consistency is an application of the consistency mapping—from user design around the life cycle wheel to the successful use of the product system by the users against all of the required scenarios.

TECHNI-KIT

During the life cycle activities, methods and tools are used to support the human being in the design or implementation of an information-intensive system. There are two parts to such support: methodology used in thinking; and tools to capture, store, and manipulate the products of thought. Both methodology and tools are used to amplify the effectiveness and productivity of the human being doing design and implementation. The methodology may be divided into

1. Theories
2. Methods (how theories are applied)
3. Criteria (for judging the development and its product)

Supporting tools may be manual (such as paper and pencils) or computer-based. We are principally interested in the computer-based tools. Such tools can provide the enhanced speed, capacity, and accuracy needed to augment the human thought process in the development of large, complex systems.

A number of methodologies and supporting tools are usually pertinent to the design and implementation of any particular information-intensive system. The application of such a collection of methodologies and tools can be made more effective if its components are selected and shaped to form an integrated, system-engineered set of methodologies and supporting tools. We call such a set a techni-kit. We believe that many of the components of a particular techni-kit can be drawn from a reservoir of techni-kit resources that are generally applicable to a class of information-intensive systems. The
generation of any particular techni-kit involves two important, but not as yet well understood, tasks: (1) the particularization of the methodologies and supporting tools for the task at hand (modify the candidate components of the techni-kit to be most effective for the particular task) and (2) the design and implementation of that particular techni-kit in an efficient and rapid manner.

Techni-kits have been proposed, and some have been built and used.\textsuperscript{10,11} To date, most of the techni-kits that the authors are aware of (including the ones that they use) are not well engineered, are incomplete, and place large clerical demands upon their users. Formalizing the concept of a techni-kit can achieve a better balance between the efforts spent on methodology and the efforts spent on tools.

CONCLUSION

The life cycle wheel model of the development of an information-intensive system has been introduced; it is based on the concept of viewpoint. In conventional life cycle models for information-intensive systems, the role of requirements analysis has been unclear. Further, such models have had difficulty in dealing with concurrent activities and iterations. By introducing the concept of a viewpoint and mappings between the results of work done from different viewpoints, these issues can be clarified. In addition, the context in which an individual works is recognized as having a major impact on the development of an information-intensive system. By adding the dimension of viewpoint to the traditional dimension of phases and system levels, a richer life cycle model has been created.

Requirements analysis is recognized as a design activity from the users' viewpoint in the extended life cycle model. Because of the increasing complexity of function and use of information-intensive systems, the insertion of a user design between user needs and implementation design allows the engineers, designers, and implementers to better handle the ever increasing levels of complexity of information-intensive systems.

The last concept introduced in this paper is that of a techni-kit. It provides a structure for considering the techniques and aids that exist or have been proposed for use in the creation of complex information-intensive systems. Methodologies and computer-based tools are two main components of any good techni-kit. They must complement one another if the techni-kit is to be an effective aid to engineers, designers, implementers, and managers.

ACKNOWLEDGMENTS

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

REFERENCES
