AD-A075 227 BATTELLE COLUMBUS LABS DURHAM N C F/G 5/2

A SURVEY OF THE STATE-OF-THE-FIELD OF REQUIREMENTS METHODOLOGY.(U)

MAY 79 D B DEVORKIN, R T OBERNDOR DAAG29-76-D-0100

UNCLASSIFIED NL

1 OF 2

AD
A075227

AIRMICS

(12)

US ARMY INSTITUTE FOR RESEARCH
IN MANAGEMENT INFORMATION AND COMPUTER SCIENCE

LEVEL



COMPUTER SYSTEMS COMMAND
INFORMATION FOR DECISION
UNITED STATES ARMY





# "A SURVEY OF THE STATE-OF-THE-FIELD OF REQUIREMENTS METHODOLOGY"

## DISTRIBUTION STATEMENT

12

Approved for public release. Distribution unlimited



This Technical Report has been reviewed and approved.

Clarence Giese, Director
U.S. Army Institute for Research
in Management Information
and Computer Science

John R. Mitchell
Chief, Computer Science
Division, U.S. Army
Institute for Research
in Management Information
and Computer Science

# A Survey of the State-of-the-Field of Requirements Methodology

May 6, 1979

by

Dr. Donald B. DeVorkin

Mr. Richard T. Oberndorf

Submitted to

Battelle Columbus Laboratories
Post Office Box 8796
Durham, North Carolina 27707

In fulfillment of DAAG29-76-D-0100

Submitted by

Advanced Technology, Inc.
591 Camino de la Reina, Suite 328
San Diego, California 92108

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | |
|---|---|
| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. |
| 3. RECIPIENT'S CATALOG NUMBER | |
| 4. TITLE (and Subtitle)<br>(6) A Survey of the State-of-the-Field of Requirements Methodology , | 5. TYPE OF REPORT & PERIOD COVERED<br>(9) Final Rept. , |
| | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>(10) Donald B. DeVorkin<br>Richard T. Oberndorf | 8. CONTRACT OR GRANT NUMBER(s)<br>(15) DAAG29-76-D-0100 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Advanced Technology, Inc.<br>591 Camino de la Reina, Suite 328<br>San Diego, California 92108 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>(11) 6 May 79 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>U.S. Army Institute for Research in Management Information and Computer Science, 313 Calculator Bldg., GIT, Atlanta, GA 30332 | 12. REPORT DATE<br>May 6, 1979 |
| | 13. NUMBER OF PAGES<br>97 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)<br>(12) 100 | 15. SECURITY CLASS. (of this report)<br>Unclassified |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE<br>N/A |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

System, Data Processing, Requirements, Methodology

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Based on a general awareness of the effects of inadequacies in requirements definition throughout the life cycle of a system, AIRMICS has initiated a comprehensive program for application of the latest methodologies in requirements analysis to the Army's information system development programs. This study is intended to provide a survey of the state-of-the-field of requirements methodology; that is, any of the component capabilities of requirements definition support such as methods, languages, and tools. The authors have (over)

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

391 859

endeavored, to the extent possible, to cite those organizations with relevant activities in requirements methodologies as developers, evaluators and users. The field is not mature and it is difficult to establish a set of desirable criteria for judging candidate methodologies. As an alternative, a small set of major factors is defined and used to determine the basic applicability of each candidate methodology to the AIRMICS program.

The authors have previous experience in the development and use of requirements definition tools and methods. On the basis of this survey effort and previous experience, they have developed a set of specific observations and recommendations in terms of concepts such as a requirements definition team, the representation of requirements for advanced systems, problem concept definition methods, resource and projection modeling, and integrated approaches to requirements definition. These observations lead to the conclusion that an appropriate methodology must take into account the Army information system environment. For this reason it is concluded that a requirements definition language should be designated only when the environment and the uses of the methodology are fully specified. At that time, the technology surveyed here will be easily transferable to the Army information systems environment.

# PREFACE

This technical report was prepared in fulfillment of Statement of Work TCN 79-075 issued to Advanced Technology, Inc. by the Battelle Columbus Laboratories under its Scientific Services Program with the U.S. Army. The technical user is the Army Institute for Research in Management Information and Computer Sciences (AIRMICS). The AIRMICS is a field activity of the U.S. Army Computer Systems Command responsible for conducting a research and technology program to improve the development and maintenance of large management information systems for personnel, logistic, and financial management applications.

The task objective is expressed in the statement of work as follows:

> The objective of this task is to have a qualified individual in the area of software requirements analysis survey the research and literature to determine which research efforts are applicable to the enclosed AIRMICS concept paper in automated management information requirements analysis.

The referenced concept paper is titled "Requirements Formulation for MIS through System Sketching". During an initial task meeting, the contracting officer's technical representative requested that the study not be limited to the system sketching technique, but address requirements analysis methodologies in general.

# TABLE OF CONTENTS


| | | Page |
|---|---|---|
| SECTION 1 | EXECUTIVE SUMMARY | 1-1 |
| SECTION 2 | PURPOSE OF THE STUDY | 2-1 |
| 2.1 | The AIRMICS Mission | 2-1 |
| 2.2 | The Requirements Problem | 2-2 |
| 2.3 | Description of the Report | 2-4 |
| 2.3.1 | Scope of the Study | 2-4 |
| 2.3.2 | Report Approach Rationale | 2-5 |
| SECTION 3 | SURVEY APPROACH | 3-1 |
| 3.1 | Requirements Activities Model | 3-1 |
| 3.1.1 | Problem Concept Definition Activity | 3-2 |
| 3.1.2 | Information Structuring Activity | 3-5 |
| 3.1.3 | Data Processing Requirements Structuring Activity | 3-9 |
| 3.1.4 | Resource and Performance Projection Activity | 3-13 |
| 3.2 | Reporting the Survey | 3-15 |
| SECTION 4 | OBSERVATIONS | 4-1 |
| 4.1 | The Requirements Definition Team | 4-1 |
| 4.2 | Representing Requirements for Advanced Systems | 4-2 |
| 4.3 | Problem Concept Definition Methods | 4-2 |
| 4.4 | Resource and Projection Modeling | 4-4 |
| 4.5 | A Perceivable Trend | 4-4 |
| 4.6 | Integrated Approaches | 4-5 |
| 4.7 | A Perspective | 4-7 |
| SECTION 5 | CONCLUSION | 5-1 |

Table of Contents cont.


| | | Page |
|---|---|---|
| APPENDIX A | ORGANIZATIONS SURVEYED | |
| A.1 | ARPA System Design Laboratory Project | A.1-1 |
| A.2 | Ballistic Missile Defense Advanced Technology Center (BMDATC) | A.2-1 |
| A.3 | Computer-Aided Design and Specification Tool (CADSAT) Project | A.3-1 |
| A.4 | Computer Sciences Corporation | A.4-1 |
| A.5 | Hughes Aircraft Company | A.5-1 |
| A.6 | Higher Order Software, Inc. | A.6-1 |
| A.7 | Integrated Computer-Aided Manufacturing (ICAM) Project | A.7-1 |
| A.8 | Information Systems Design Optimization System (ISDOS) Project | A.8-1 |
| A.9 | Martin Marietta Aerospace Company | A.9-1 |
| A.10 | Naval Weapons Center, China Lake | A.10-1 |
| A.11 | Softech, Inc. | A.11-1 |
| A.12 | Sperry-Univac, Inc. | A.12-1 |
| A.13 | System Development Corporation (SDC) | A.13-1 |
| A.14 | Teledyne Brown Engineering, Inc. | A.14-1 |
| A.15 | Xerox Corporation | A.15-1 |
| APPENDIX B | BIBLIOGRAPHY | B-1 |

## LIST OF FIGURES


| Figure | | Page |
|---|---|---|
| 3.1 | Basic Requirements Activity Iteration Model | 3-3 |
| 3.2 | Matured Requirements Activity Iteration Model | 3-8 |
| 3.3 | Requirements-Design Iteration Model | 3-14 |

## SECTION 1. EXECUTIVE SUMMARY

Based on a general awareness of the effects of inadequacies in requirements definition throughout the life cycle of a system, AIRMICS has initiated a comprehensive program for application of the latest methodologies in requirements analysis to the Army's information system development programs. This study is intended to provide a survey of the state-of-the-field of requirements methodology; that is, any of the component capabilities of requirements definition support such as methods, languages, and tools. The authors have endeavored, to the extent possible, to cite those organizations with relevant activities in requirements methodologies as developers, evaluators and users. The field is not mature and it is difficult to establish a set of desirable criteria for judging candidate methodologies. As an alternative, a small set of major factors is defined and used to determine the basic applicability of each candidate methodology to the AIRMICS program.

The authors have previous experience in the development and use of requirements definition tools and methods. On the basis of this survey effort and previous experience, they have developed a set of specific observations and recommendations in terms of concepts such as a requirements definition team, the representation of requirements for advanced systems, problem concept definition methods, resource and projection modeling, and integrated approaches to requirements definition. These observations lead to the conclusion that an appropriate methodology must take into account the Army information system environment. For this reason it is concluded that a requirements definition language should be designated only when the environment and the uses of the methodology are fully specified. At that time, the technology surveyed here will be easily transferable to the Army information systems environment.

# SECTION 2. PURPOSE OF THE STUDY

## 2.1 THE AIRMICS MISSION

The Army Institute for Research in Management Information and Computer Sciences (AIRMICS) has research responsibilities across the whole of the Army information system needs. The existing Standard Army Management Information Systems (STAMIS) is example of:

1. The type and scope of the information system applications.
2. The products resulting from the Army information system development/modification environment, an environment which consists of the proponent agencies and their constituents, together with the Computer Systems Command (CSC) acting as developer.*

The CSC being responsible for the non-tactical ADP needs of the Army has, most naturally, traditionally reflected the thinking and approaches of the business processing community at large.

Certain factors can be perceived which together are producing a need for improvement in present CSC approaches and capability for ADP requirements support. In particular, we note:

- The great need to reduce the cost of software development and maintenance.
- Awareness that systems too often fall short of a level and type of service which should have been accomplished.
- New emphasis on MIS support. The overall CSC responsibility is becoming more complex.

---

*The terms "users" and developers" are used in two ways in this report. In the first usage, such as is the case here, "users" refers to that group which is the ultimate beneficiary of a given system, and "developers" are those who produce the system. In the Army environment, these would ordinarily be the proponent agency and the CSC respectively. The second usage of these terms refers to the users and developers of a particular requirements methodology. In this case, "users" of a methodology might be the Army CSC, and "developers" might be, for example, the University of Michigan (the developers of PSL/PSA). It will always be clear from context which usage of these terms applies.

- Increasingly complex technical issues with respect to networking, security, deployability, and in some cases, mobility.

There is another, and most important, factor which must be considered as the CSC moves to meet present and next generation challenges. The ADP community is now fractionated by the rapidity and diversity of technology advances. A general concensus as to the right way to go forward is no longer available for the ADP community.

The preceding considerations lead to the conclusion that the Army CSC must be able to rely on AIRMICS to:

1. Chart an appropriate course through the variety of hardware alternatives.
2. Take advantage of emerging software/systems disciplines.
3. Provide and recommend system development techniques.

The proponent agencies must be able to rely on AIRMICS to perform the research necessary to obtain:

1. More effective means to state and manage their requirements.
2. The framework for quantitatively improving the quality of the Army information systems.

To work toward these goals, AIRMICS has initiated a comprehensive, phased program. The first phase of the program deals with the problem of requirements definition. It is the natural starting place. Accomplishments of this phase will serve dual purposes. Capability will be obtained such that:

1. AIRMICS will be able to ensure the correctness of its own requirements; and hence, the appropriateness of the direction to be followed for the remaining program phases.

2. The present and near-term ADP problems of the Army can be correctly defined.

## 2.2 THE REQUIREMENTS PROBLEM

The current AIRMICS requirements definition methodology phase coincides with a growing general awareness of the effects of inadequacies in requirements definition throughout the life cycle of a system. This awareness is now reflected in the major policy centers of the DOD, in the academic community, and within industry.

The survey by Ramamoorthy and So [1977] describes the typical situation:

> Traditional means to analyze and state requirements result in unsatisfactory specifications. The following figures indicate the magnitude of the problems: manual examination of the requirement specification reveals that there are more than 50 problems in a specification document of about 48 double spaced typewritten pages and 972 problems in a specification containing 8248 requirements and support paragraphs (2500 pages). Most frequently occuring problems are (1) incorrect requirements, and (2) inconsistent and incompatible requirements, and (3) requirements unclear.

A very clear insight is provided in the FY 79-83 R & D Technology Plan by the R & D Technology Panel to the Management Steering Committee for Embedded Computer Resources (MSC-ECR) of DDR&E. We quote directly:

I. Problem/Issue Summary

- Dissatisfied Users
- Excessive Requirements
- Incomplete Requirements
- Inconsistent Requirements
- Untestable Requirements
- Untraceable Requirements
- Infeasible Requirements

II. Research Direction and Action

Requirements specification is, and always will be a problem. The process involves the transformation and synthesis of thoughts, desires, needs, fears, and perceptions into a precise specification of the problem the system is to solve. We will always be faced with conflicting, inconsistent, and changeable missions, and responses. It is important to recognize that we cannot technologically eliminate this fact, and instead to concentrate on providing rapid insight into the technical implications of stated system requirements on computer resources (and vice versa), identifying risk areas, and exploring implementation alternatives iteratively before making hardware, schedule, and projected cost commitments. The user needs to have the closed-loop opportunity to see and feel the implications of his requirements before he becomes irrevocably committed

> to them. Our primary research thrust in the near-term, then, is aimed at "Software First" Technology (in conjunction with emulation facilities testbeds); Competitive Prototype Guidelines and Evaluation Criteria; and Requirement Trace Tools. Tools for Semi-Automated Consistency and Completeness Analysis and Requirements Decomposition are promising basic research thrusts.

The R & D Technology Panel FY 79-83 Technology Plan made recommendations to the MSC-ECR which adopted those recommendations as objectives for overcoming deficiencies in the technology base for both embedded and general purpose computer application areas. The objective with respect to requirements definition is again incisive and to the point. We cite:

> "Develop methods, languages and tools for describing and validating requirements."

It is the purpose of this study to support the objective of the R & D Technology Panel tailored to the specific needs of AIRMICS by providing a baseline of the state-of-the-field as it relates to the AIRMICS mission.

## 2.3 DESCRIPTION OF THE REPORT

### 2.3.1 Scope of the Study

The word methodology is used here in a general sense to refer to any of the component capabilities of requirements definition support: methods, languages, and tools. The study, then, is a survey of the state-of-the-field of requirements methodology. It is not however, the type of survey found in the journals.

There are already many valuable contributions to the requirements methodology survey literature. We mention, just as example, the work of Ramamoorthy and So [1978], Taggart and Tharp [1977].

Our purposes here are very much different from that of a survey to a general audience. It is the explicit intent of the study to provide AIRMICS with conclusions and recommendations such that AIRMICS can move its own program forward in significant ways. We are therefore, much less concerned with describing methodology than would be the case for a journal survey. Instead, the study is focused on reporting candidate methodology based upon:

- Utility
- Near-term developments expected
- Availability

- Credentials
- Relationship to the life-cycle

With this orientation, the study has endeavored, to the extent possible, to bring attention to those organizations and projects which are involved with relevant methodology activities either as:

- Developers
- Evaluators
- Users

As will be seen, the activities of a group may span these categories.

The field is not mature. Therefore it cannot be supposed that it is meaningful to set out a list of desirable features and then judge how well a candidate fulfills those desirable qualities. Instead, a small set of major factors will be defined. These factors will be used to determine the basic applicability of candidate methodology to the AIRMICS program.

The study is focused on major requirements methodology thrusts, based on the material that could be obtained within the scope of the effort. Concepts discussed in the literature, but not yet sufficiently practiced, are not reported. Certain papers are referenced for purposes of conceptual framework, but no attempt has been made to do a paper by paper evaluation. The authors do, however, feel that the scope of the study is sufficient to support the conclusions reached, and the recommendations put forward.

### 2.3.2 Report Approach Rationale

The study authors have previous participation in the development of requirements methodology. The present effort together with the experience of the authors has yielded some clear conclusions and recommendations. These results, however, should be readily apparent to any experienced observer without biased or vested interest. Finely drawn analysis on survey results is not needed at this time. This is fortunate since such analysis would not be supportable owing to the state-of-the-field.

The next section, Section 3, sets up the framework for the survey and the techniques to be used. Section 4 provides some observations and recommendations which serve to support the concluding remarks of Section 5. Discussion of specific methodologies surveyed is placed in Appendix A. Appendix A is principally structured according to the organization involved rather than by methodology type categories. This reflects both the data gathering approach and an emphasis on organizational support.

# SECTION 3. SURVEY APPROACH

## 3.1 REQUIREMENTS ACTIVITIES MODEL

In order to survey requirements methodology and evaluate findings, we must first understand the nature of the activities which go into the requirements definition process.

A requirements methodology must support the activities described below. In each case the activity results in description/documentation. Therefore language (in a general sense) is crucial to the endeavor of the activity.

Analysis of various kinds are needed as the means whereby the description provided by an activity is validated, and features of importance are calculated and displayed.

The requirements definition process is fundamentally iterative in nature. Lack of appropriate methodology and requirements engineering tools account for much of the noticeable failure of development projects to adequately produce and maintain the system and subsystem requirements definitions. We therefore have need for methodology which, in addition to supporting the individual activities of the requirements definition process, is cognizant of and is also supportive of the iterative aspect of the problem.

Opinion among experts differs with respect to the interactions among the various pursuits involved in a major system/software requirements definition. The approach here has been to limit consideration to major activities and to formulate as simple a model of the iterative interactions as is consistent with providing a common basis for discussing the surveyed methodologies.

The model reflects the concensus that there is a distinction between problem requirements and solution-system requirements. In practice, this distinction is often not maintained. System requirements are often specified in lieu of a formal problem definition.

There is another concensus in the field: requirements must be specified hierarchically. However, hierarchy has different meanings to different people. This is not surprising since there are indeed different hierarchically organized pursuits encompassed by the requirements activities.

In reality, the relationships among the major activities of the requirements definition process are not fixed. As the requirements process progresses, the relationships and focus among the activities evolve. We have illustrated this evolution by a series of three models representing the relationships among the major requirements definition activities. Figure 3.1 shows the Basic Requirements Activity Iteration Model. The following discusses the individual activities and the relationships among them.

### 3.1.1 Problem Concept Definition Activity

This activity is generally informal. Documentation is usually English language oriented.

Uhrig [1978] has pointed out that any such activity provides a level of explicit requirements knowledge and leaves the next level of requirements particulars as implicit. To formulate requirements is to express tacit knowledge and awareness. Uhrig states:

> "One may fairly claim that the central problem of system requirements is to address this tacit dimension."

There is an altogether different aspect of the problem definition. It is that sometimes we must proceed with only certain portions of the problem definable to any meaningful level. When there is a feeling that the portion we can make explicit represents "enough" of the problem, we are willing to move ahead and believe we can add the rest later. "Enough" here usually has to do with the resources ultimately to be required. Providing a safety factor in resource estimates is a familiar procedure.

There is an important point. It is that we want to have assurance that when unknowable portions of the problem can be made knowable, the "hard" requirements previously defined are not materially impacted. That is, we do not want to have what is not yet definable to altogether undo the results of efforts based on partial knowledge.

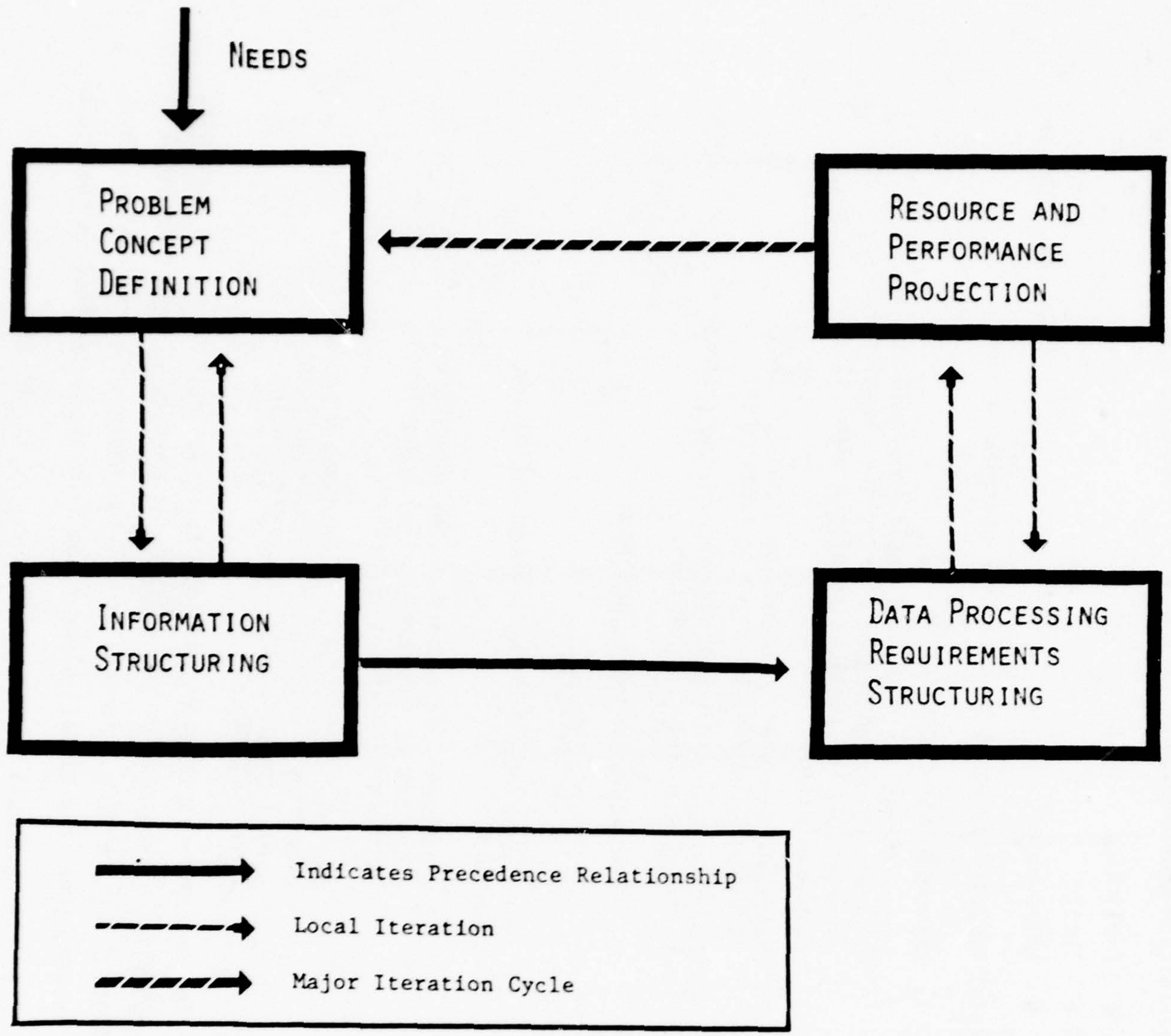


Figure 3.1. Basic Requirements Activity Iteration Model

At the problem concept stage, we would like to be able to partition the problem definition knowledge into the following categories:

- Explicit
- Tacit
- Unknowable

A useful methodology would be one which contributes to:

1. Documentation organization.
2. Ensuring that the domain of tacit knowledge is entirely relegated to the hierarchical level representing the particulars of the explicit knowledge level.
3. Rules and measures to bound the impact of that portion of the problem which is unknowable.
4. Means to move and track portions of the problem definition as they change from:

    a. Unknowable to tacit, and from

    b. Tacit to explicit.

5. Procedures to expand the problem definition because:

    a. When tacit particulars are made explicit, a new set of (hierarchically) lower level tacit particulars is generated.

    b. The specifics of what is unknowable become more finely definable. This establishes a need for documentation extension and modification.

So far, we have discussed methodological issues related to the mechanics of the activity. There remains the question of methods for the conceptual perception of the needs/requirements underlying the problem definition. By way of answer, there are certain clear rules from the experience in the field:

1. The eventual users must be surveyed to obtain fundamental problem needs/requirements.
2. Users and developers must be teamed early.

3. Users must remain in the loop during the iterative requirements definition process.

Certain elaborations on these rules are necessary. First, we draw a distinction between hands-on users and user management. Then with respect to Rule 1, the user survey should encompass a large sampling of both types of users. A small working team of users and developers is to be interfaced according to Rule 2. The first task of the team is to evaluate the user needs survey. It is clear that it would be desirable to have had the same team for the formulation and conduct of the survey.

It should be pointed out that the eventual system developer is seldom on board at such an early point due to procurement philosophy. This specific situation will be unalterable. The simple reason is that the job to be bid is not yet defined.

In view of this last consideration, we must substitute the words systems engineer or technologist for developer in the Rules. There are however, problems with this. The difficulties center on the ability of the team to make meaningful feasibility and cost vs. performance vs. schedule constraint judgements. A recommendation for the situation will be made in the next section.

Finally, with respect to Rule 3, the team must become structured so that only management-level operational and financial decision issues are put before the user management team members. This is not altogether in the interests of efficiency. It is rather the usual consideration of decision made at the most qualified level.

### 3.1.2 Information Structuring Activity

This activity identifies information production, use and flow. The result is a formal definition of the problem as a set of information transformations and relationships together with relevant parameters. The parameterized information characteristics would include absolute or relative values having to do with:

- Access
- Accuracy
- Range
- Age
- Stimulus
- Security
- Duration of information maintenance
- Source

In at least the early stages of the requirements definition phase, the Information Structuring Activity iterates directly with the Problem Concept Definition. It therefore reflects the current partitions of knowledge about the problem. As discussed in the previous subsection, the explicit and tacit knowledge partitions are hierarchically related.

The more usual notion of hierarchy results from the information structure obtained by information decomposition. But any approach to the structuring of information through decomposition rests on the organizational schema pursued. For example, the information could be structured according to type of material described, or by relevance to an output report, or based on source agency, etc. Decompositions would refine details, but would be conducted in line with the structuring schema. There is no one way to structure; there is no general concensus in the field. However, there is <u>some</u> agreement that it is desirable to:

1. Define the information (not data) report outputs desired, and
2. Work backwards, upstream so to speak, against the information transformation flow.

The Information Structuring Activity is <u>problem</u> definition oriented. The results are implementation - and as much as possible, system design - independent. The study has tried to emphasize this activity because it is especially important to the CSC applications. This is in distinction to an arena such as weapon system processing. For that set of applications, it is very often the case that no information structuring is meaningful without consideration of the physical realities of the sensors, the platform, and the weapons to be available. This is because the physical devices themselves set much of the ultimate processing needs.

Very often then, we find that for weapon systems, after only a slight concept definition effort, solution-system requirements are pursued. The degree to which this approach is commendable or is counter productive for weapon system development is not at issue here. The subject has been brought up for contrast. There is very little to "hang ones hat on" in an information system problem except what the users say they want and the source data availability. Therefore, the Information Structuring Activity is of major significance.

If the methodology of the structuring is at all appropriate, the study authors observe that the <u>structured</u> informational definition of the problem becomes much easier to work with than the Problem Concept Definition documentation. This seems true even when the methodology is informal, but practiced by experienced requirements analysts. The reasons have to do with:

1. The brevity obtained; the ability to display material in digestible form.
2. The appropriateness of the language (vs. English).
3. The suggestiveness the structural organization provides.

For these reasons, after enough of the problem is understood, further problem definition gravitates to a direct use of the structured informational definition of the problem. While this may not become completely so, it is sufficiently the case such that a second-stage Requirements Activity Iteration Model is necessary. This model is shown in Figure 3.2.

Methods for establishing an appropriate information structural schema are central to this activity. Rules and procedures for decomposition are equally important. Any methodology operating in this area should be capable of materially supporting consistency and completeness analysis. The support may rely on either manual or automated means, but the analysis must be featured.

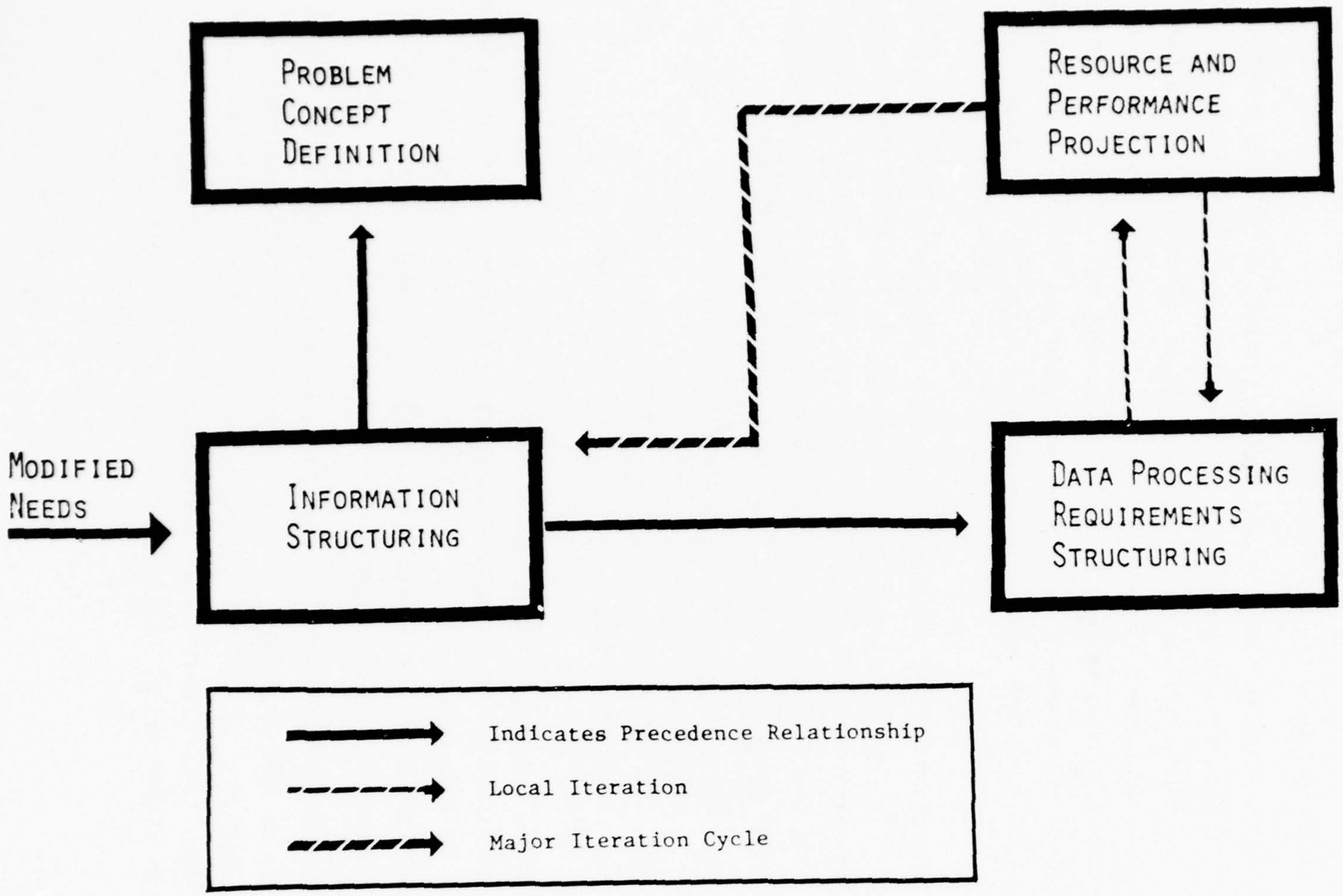


Figure 3.2. Matured Requirements Activity Iteration Model

Problem oriented languages are generally being employed in support of the Information Structuring Activity. Diagramatic approaches are common. Graphics are used as a visual aid in conveying the structural relationships. In some cases we find the diagramming techniques matured and formalized to the point of being the major facility of the problem definition language.

In some cases, the problem definition languages, are interactively machine supported. This includes the graphical types, but of course on a less generally available basis. For one thing, interactive machine support for diagramatic structuring requires a graphics terminal. While such terminals are available in the marketplace, they are still not commonly provided in general purpose situations.

In conjunction with automated language, a database management system and report generator are obvious, necessary tools.

3.1.3 Data Processing Requirements Structuring Activity

This activity deals with the structuring of requirements for the solution system. For both the model of Figure 3.1 and for the matured model of Figure 3.2, the source for the data processing (DP) requirements is the structured information problem definition.

Determining solution system requirements implies a system concept defined to the level at which DP requirements are desired. In this sense, DP system requirements cannot be independent of the hardware/software technology base, and must reflect the architecture of the system concept.

The journal literature seems confused on this last point. The literature stresses implementation independent requirements definition. This is as it should be; a definition of system requirements should be free of implementation specific details. But a complete disinvolvement from the system concept, system architecture, and technology capabilities/limitations, would place the resulting requirements definition back in the problem definition camp.

A further general confusion has to do with the design aspect of actual DP system requirements definitions. The literature goes to some lengths to draw fine distinctions between system requirements definition and design. Such distinctions are potentially counterproductive. This is because the aims and impacts of system requirements definition are very much different in practice from those envisioned in current theory and abstraction.

Any DP solution-system specification must fulfill the information production requirements specified by the Information Structuring Activity. Additionally, there may be other requirements with respect to an actual system. For example:

- The system may have to meet interface and communications standards imposed by other in-place systems or equipments.
- The system configuration may be required to preserve previous software investments.
- Policy implication on system failure stated as back-up and recovery requirements must be included.
- Size, weight, etc. may be of enough concern to require definition of allowable values.

We see that the DP Requirements Structuring Activity:

1. Translates and incorporates the structured informational definition of the problem into a DP system context, and
2. Factors in the pre-conditions and constraints on a solution-system.

The degree of system requirements definition activity which does not directly translate from the informational aspects of the problem is, in general, very considerable. For example, consider the need to define requirements for:

1. A user language(s) and programming tools.
2. Interactive prompting, menu selection; the capabilities of the terminal itself.
3. Database management.
4. Operating system features and performance (efficiency) in view of life cycle concerns such as modifiability and extensibility.

5. Diagnostics at all levels.
6. User accountability; i. e. the nature of the internal (software) system which monitors the way system resource is used and accounted for.

The DP system requirements definition is to contain all knowable, information relevant to a solution system. The issues which remain open to tradeoffs and choice are the domain of the Design Activity. It is clear that the level of specificity of DP system requirements definition is entirely dependent on the particular situation.

We make the observation that a system is not built from requirements; it is built from design. Structuring of system requirements has as its real purpose:

1. Verification. System requirements are the foundation for test plans. Systems are built to (satisfy) requirements.
2. An instrument for management decision and organization.

Technical literature supports our second point. For example, a paper by Giese [1979] deals with system requirements partitioning (an approach to structuring) within the context of high level system architecture. Yet it has as its basic motivation and problem areas treated, identification of contractually coherent subsystems and administration of interfaces, which are primarily management issues.

This perspective helps set the stage for explanation of the role of design in the DP Requirements Structuring Activity. DP system requirements definition and design are mutually contributory, and mutually supportive pursuits. It is altogether reasonable that DP system requirements definition makes us of, incorporates, or becomes merged with preliminary design. The authors note that methodologists not withstanding, the practical realities have made, and will keep this the case.

As an extension of the concepts found in Uhrig's paper, it is suggested that preliminary design serves to make explicit the implicit awareness about the solution system known at the start of the system requirements definition process. The notion goes further and adds the idea that the explicit knowledge of system requirements undergoes modification as the implicit particulars of preliminary design are articulated.

The process is iterative. It can even be that at some point, a certain system requirements knowledge is carried only implicitly and must be made explicit to articulate particulars for purposes of advancing design. We now have the nature of the relationship of preliminary design to system requirements definition.

For a serious methodological approach, it will be assumed that an Information Structuring Activity has produced a structured informational definition of the problem. To illustrate a point, let us suppose that then someone were to step forward and offer an already available system. Our job, suddenly, is to validate the performance of the system against the problem definition. If we have reason to suppose the applicability of the system, we have no need to back up and ask for solution-system requirements definition and design.

It is often the case that we can deduce a good deal about a system which will meet the problem requirements and be within the major constraints of schedule, money, compatibility, etc. There is good reason for being able to take this procedural jump. It is because in attempting to supply a DP system, or a design of a DP system, we only seek (in practice) sufficiency. At most we obtain a few sufficient alternatives and pick the "best" one.

When pursuing a system requirements definition on-the-other-hand, we attempt a balance. We do not want to either under or over require the system. We are therefore put in the situation of having to do requirements tuning without knowing enough to properly do so. The result is a difficult, consuming activity of (later) doubtful utility.

It is for the above reasons that we do not formally list a Conceptual System Requirements Activity in the model of Figure 3.1. To do so would be counter to the realities of the practicing field. The authors have in fact found no methodology available to address such an activity.

We will suppose in our model, that a conceptual system definition is folded into the DP Requirements Structuring Activity as the architectural rationale for the structuring schema and decomposition rules. Procedures for decomposition again are relevant and come under the heading of methods.

For the DP Requirements Structuring Activity, we expect language support to reflect the proximity of the activity to design. Therefore, in the field we can expect a convenient design language to be borrowed for purposes of the DP Requirements Structuring Activity.

Some design languages are basically graphical representations. Again, the remarks made in the previous subsection with respect to machine support for language are applicable. DBMSs, report generators and (possibly) interactive display are all valuable supporting capabilities.

3.1.4 Resource and Performance Projection Activity

This activity is initiated in the requirements phase, and continues throughout development. As project development progresses, more reliable estimates become available. The results of the activity are central to both project level and technical level decision making. An activity iteration model which shows this and incorporates a preliminary design activity is shown in Figure 3.3.

The basic tools in support of resource and performance projection are modeling and simulation. During early stages, manual means may be sufficient. There is an operational concensus in the field that for projects of any size, automated simulation is necessary as support to design and implementation.

Language for developing a simulator may be nothing other than one of the general purpose simulation languages such as GPSS or SIMULA. On-the-other-hand with appropriate automated support to the other activities, it has been demonstrated that simulation can be obtained directly from an automated requirements database.

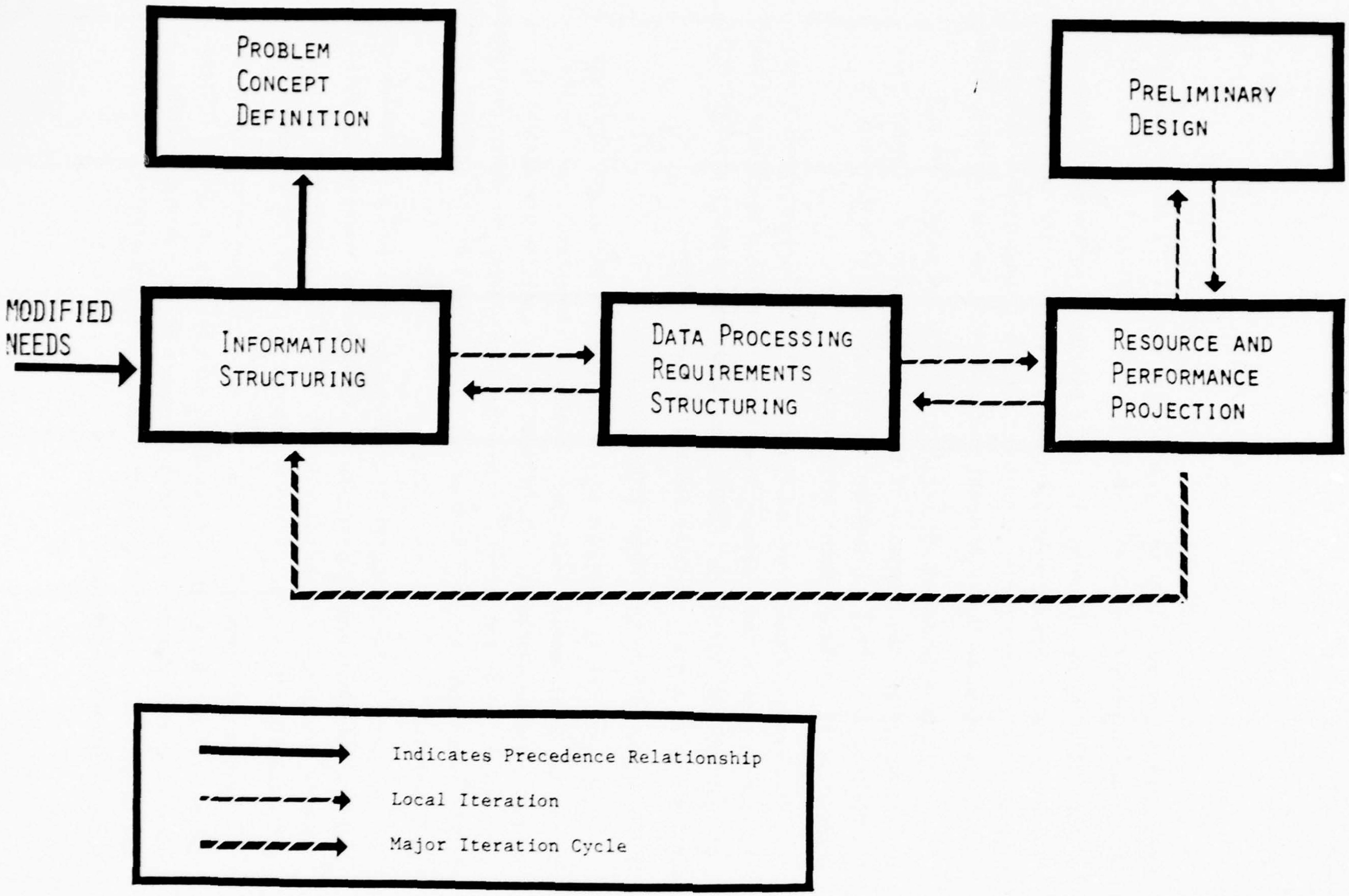


Figure 3.3. Requirements - Design Iteration Model

There is not much available in the field by way of general methods and rules of procedure for resource and performance projection. Experience seems to be the only real guide. Pursuit of this activity tends to be as application dependent as the degree of awareness about a solution-system permits. Traditionally, there has been little transference of techniques or tools from one application to another.

The possibility of automatic simulation generation directly from an automated requirements/design database is a promising avenue to relieve the situation. There remains open however, the issue of firm methodology to support obtaining meaningful parameter estimates to use in the simulations and projection models.

## 3.2 REPORTING THE SURVEY

The study has attempted identification of extant methodology which can be adopted/adapted for the AIRMICS mission. The approach has been to survey certain of those organizations involved in major ways with methodology either as developers, sponsors, evaluators/integrators, or users.

The study has primarily sought to learn:

1. Which requirements definition activities are addressed, and the type of support provided.
2. The technical and organizational context for the requirements definition methodology produced.
3. Where the methodology is heading and its usefulness to AIRMICS.
4. Whether there are credentials which make the methodology real.

Individual reports for each surveyed organization are presented in Appendix A. Each report is comprised of:

1. A one-page METHODOLOGY DATA SUMMARY FORM as an overview.
2. A structured set of findings and results termed a METHODOLOGY REPORT.

These reports represent the basic data which, together with the authors' experience, are the basis for the observations and recommendations made in the next two sections.

# SECTION 4. OBSERVATIONS

Based on information obtained in conducting this survey, and the experience of the authors in the field of requirements methodology, this section presents a set of observations and related recommendations for consideration by AIRMICS. Because these observations are the result of a synthesis of the findings of this study, they are not presented in any particular order with respect to other sections of this report. The recommendations stated here, and those in Appendix A, are intended to be supportive of the conclusion drawn in Section 5.

## 4.1 THE REQUIREMENTS DEFINITION TEAM

There are problems with the Requirements Definition Team approach. As previously mentioned, the system developer is rarely on board to participate in the problem and system requirements definition activities. Further, users who have been initially surveyed for needs, are ordinarily not available as ongoing requirements compromise and modification occurs. Requirements team members from the user community are not necessarily able to represent interests outside of their own particular areas.

For these reasons it is suggested that a third party be included in the composition of the requirements team. The third party would be an independent, non-vested, systems oriented house.

In the requirements phase, the third party would provide technology support to the requirements team. Here, the role would be similar to that of the developer.

The third party has another duty during the requirements phase. It is to understand the issues of the users. When the developer comes on board, the third party switches roles to now represent the user community. It is the third party which can provide:

1. The corporate memory.
2. The view from both sides of the fence.

It is interesting to propose that the third party take the structured informational problem definition and the system requirements specification back to sampling of the user community before proceeding to detailed design and implementation.

The user community is responsible for the needs leading to a problem definition. From that time forward, there are management and technology based trade-offs, compromises, decisions, and preferences. A user group of some size should be consulted in a buy-off of the system requirements.

Further benefit from the corporate memory investment in the third party can be obtained in later development phases. The third party would be the natural agent for the formulation of integration and test plans.

## 4.2 REPRESENTING REQUIREMENTS FOR ADVANCED SYSTEMS

Diagramatic requirements and design representations are popular despite the investments necessary to properly support the approach. This is because of the basic human propensity toward visual information. We can, however, fall victim to our own talents.

It must first be emphasized that graphics have indeed been found very helpful for purposes of both data and functional decomposition. There are, however, many other kinds of issues having to do with the nature of systems capitalizing on current or near term technology. By way of example, we ask: how does one diagram <u>requirements</u> related to:

- Intersystem communications protocol.
- Distributed system integration.
- Asynchronous pre-emption.
- Adaptive resource management.
- Prompting at a man-machine interface; especially when there are potentially a great many possible choices of response in a non-fixed sequence of interactions.
- Database redundancy.
- Reconfiguration.

We have learned about the representation of requirements which can be deduced from decomposition. We now need to turn R&D attention to the nature of requirements which are obtained by synthesis (those which come about only because a combination of other requirements are operative).

## 4.3 PROBLEM CONCEPT DEFINITION METHODS

Methods related to Problem Concept Definition are often the proprietary product of MIS consulting firms. The Structured Analysis and Design Technique (SADT) product of Softech is the leading example. Details of approach are therefore not often available. Even so, certain things are clear with respect to the definition of general business and MIS processing problems:

1. The user community at large should be consulted.
2. A balanced requirements definition team should be formed and supported.

It is the feeling of the authors that obtaining an appropriate Problem Concept Definition is basically a direct management responsibility, whether the problem lies in information processing or some other area. In fact, this would seem to be the major purpose of management. For this purpose, management has long been in the business of putting together qualified teams and soliciting data and opinions from affected parties.

Problem-definition problem solving techniques are openly taught across the whole of management science. It is strongly recommended that AIRMICS not become mired in the perpetuated mystery given to MIS problem definition.

It is proposed that a tiger team could be formed in an expeditious manner to produce detailed Problem Concept Definition methods. The team would consist of:

1. Management consultants experienced in survey techniques, reporting techniques, etc.
2. Experts on CSC organization, responsibilities, and in-place systems.
3. Users.
4. Technologists.

A newly formed project is underway in the Navy which has a problem area of direct interest to AIRMICS. The project is the establishment of a central design agent (CDA) in the Naval Supply Systems Command to develop, implement, and maintain a uniform Naval Industrially Funded (NIF) financial ADP system for all Research, Development, Test and Evaluation (RDT&E) activities.

The CDA is currently engaged in requirements definition. It is suggested that AIRMICS be cognizant of approaches taken and problems encountered.

4.4 RESOURCE AND PROJECTION MODELING

System resource and performance projection capability must be considered a major AIRMICS product goal. As discussed above, except for general purpose simulation languages, interdisciplinary, transferrable methodology for this purpose is lacking.

The authors observe that for much of the problem domain of CSC responsibility, machine aided model analysis may be a more meaningful approach than transactional simulation. The models would be oriented specifically to the task of system resource and performance projection in a changing mission and technology environment. It is expected that numerical solutions would be obtained through recursive techniques.

It is therefore recommended that AIRMICS baseline:

1. The level of modeling and simulation needs.
2. The utility of available models.
3. The most promising numerical mathematical approaches.

4.5 A PERCEIVABLE TREND

There is at this time a momentum toward the use of the Problem Statement Language/Problem Statement Analyzer (PSL/PSA) and its derivative version User Requirements Language/User Requirements Analyzer (URL/URA).

The Navy is engaged in PSL/PSA support, evaluation, and dissemination through the Naval Ocean Systems Center (NOSC) System Design Laboratory (SDL) and other activities. SDL is reported on in the Appendix. The Navy, on both coasts, is actively using PSL/PSA for requirements and design documentation. The authors are aware of this type of activity in relation to:

1. The combat systems of the nuclear guided missile cruiser, CGN 42.
2. The combat systems of the AEGIS guided missile destroyer, DDG 47.
3. A NAVMMACPAC resource and manpower planning system.

4. Real-time range control for the Electronic Warfare Threat Environment Simulation Range (ECHO Range) at the Naval Weapons Center (NWC).

5. The collision avoidance navigation system (HICANS) for high speed surface craft.

Further, only very recently, the newly formed Combat Systems Architecture (CSA) Program Office has mandated that PSL/PSA documentation must support CSA requirements and design. The CSA office is chartered to have a signigicant impact on the weapons system configurations and performance for the next generation of major surface conbatants. The posture of the CSA office is expected to promote the use of PSL/PSA in many areas of Navy R&D. It is also important to note that the resulting PSL/PSA documentation will be standarized.

The Air Force has developed the PSL/PSA derivative, URL/URA, to increase capability. URL/URA is in current use by Air Force projects which include:

- SEEKFROST
- Joint Surveillance System (JSS)
- E3A/AWACS System
- Integrated Computer-Aided Manufacturing (ICAM) Project

Initially, PSL/PSA was intended for commercial applications. As example, the Chase Manhattan Bank and AT&T Long Lines have each been major users. Both organizations have experience with PSL/PSA over many projects.

It is recommended that AIRMICS:

1. Note and weigh the current trend.
2. Investigate the experience of commercial information systems users since these problem areas are directly applicable to AIRMICS.

## 4.6 INTEGRATED APPROACHES

A number of the organizations surveyed have, or are in the process of, developing an integrated approach to a battery of tools to support the requirements (and design) activities. Where this has been found to be the case, the individual report contained in Appendix A will recommend AIRMICS attention to the effort. An across the board integrated approach is not the only reason to suggest interest. Individual components of utility are also noted. However, it is important for AIRMICS to be cognizant of those efforts of the scope at least comparable to the magnitude of the Army information system problem.

The components of an automated system of this kind are, in general terms:

1. Input language.
2. DBMS.
3. Static Analyzer.
4. Simulation generator.

It has begun to be realized that there are very different, valuable, orientations from which to define requirements. In particular, there is a process vs. data flow distinction to be drawn. The orientation(s) from which to define requirements should be natural to the issues of the application area.

The most comfortable and productive orientation by which humans can define requirements in an applications area is not necessarily an appropriate organization from which to produce analysis. For this reason, the starting place for an automated, integrated requirements definition system of tools is to define:

1. The information content of the automated requirements data base.
2. The static analysis to be available.

From this, one proceeds to design of a data base schema and requirements for a DBMS. If a (more or less) appropriate DBMS is available, an initial, and integratable, capability is easily obtainable. This explains the trend toward use of PSL/PSA. The requirements definition language, PSL, may or may not be completely appropriate to an application; but the data management and static analysis capabilities of PSA have been found useful.

Once the DBMS and data base schema are in place, and a preferred input language orientation is known, a translator is developed as an integration mechanism. The requirements definition input language can, of course, be diagramatic so long as it is machine supported.

It can even be that more than one orientation to requirements definition is desired. In this case, several input languages and their respective translators must be made available. Consistency can be enforced through the DBMS and analysis capabilities.

There is no general agreement in the field with respect to requirements definition language. The orientation, and even the kind of specification of details to be supported depends on individual conceptions.

If the data base schema and DBMS have been cooperatively devised with respect to simulation, then automatic simulation generation is readily obtained. Even if this is not the case, formatted comment entries have been found in practice sufficient to serve as the means for specifying dynamic parameters.

The real issue for the Resource and Performance Projection Activity is in defining the type of model(s). More than one model may be used to provide specific insights. The static analysis capability can be used to insure consistant input sets.

### 4.7 A PERSPECTIVE

It is rather easy to get carried away with respect to methodology in general, and with respect to automated tools in particular. There is a desire for tools to be exactly appropriate, and to cover all aspects of a requirements definition endeavor.

We can often go a long way on just a reasonable amount of support. By way of analogy, let us consider automated support for document production. To set the context, suppose that we are a systems house responding to an RFP with a slam bang proposal effort. Experts contribute to specific subsections; review teams call for re-writes. Everybody uses the automated document production support facility and the proposal evolves. The tools are purposefully simple so that everyone can use them. Certain nice capabilities are not provided. These might include: right justification, extended symbol set, automatic Table of Contents generation, etc. But the primary objectives are accomplished. Many people can simultaneously contribute in a coordinated way; and, the physical burden to revision is relieved.

Whatever the tools cannot support, can be handled by the people, and can be tolerated in comparison with a no tool environment.

The authors feel that it is important for AIRMICS to be oriented toward methodology objectives to specifically:

- Reduce rote drudgery, and
- Promote consistancy of communication and coordination among all those involved,

but most importantly to:

- Provide that level of analysis not available without machine support.

# SECTION 5. CONCLUSION

It is the conclusion of this study that the state-of-the-field of requirements definition is such that existing tools and methodology will be easily adaptable to the Army environment once this environment has been carefully defined. This adaptation, however, will not be appropriate unless it supports the process by which the Army generates its information system requirements and increments implementations.

It will be a necessary condition that, owing to the scope of Army information systems, the requirements methodology will be supported by automated tools. To develop automated requirements data base(s), machine processable requirements definition language must be designated. The language and methodology would serve as the technical means of communication within and between proponent agencies and the CSC. The Army information systems baselines and the performance and resource-utilization projection tools then provide the means for benefit-to-cost determinations for modifications. The analysis tools form the basis for quality assurance with respect to requirements for fielded systems.

Improvement of the quality of Army information systems cannot be obtained without improvements to the processes generating these systems. The methodology mentioned above represents an obtainable means to significantly strengthen the process by which these systems are developed and maintained. To provide such a methodology and insure its appropriateness, a specific set of objectives must be accomplished in the following order:

1. Define the analysis to be performed. The types of requirements errors and problems presently evidenced by the fielded systems can serve as a guide to the analysis to be performed once Army information systems requirements are baselined in machine supported form. The definition must include the information which is to be supplied to the analysis routines as well as the output results.
2. Define the types of simulation to be performed and the information necessary as input.

3. Determine who will use the language and requirements baselines for communication. Determine what will be communicated.

4. Use the results of 1, 2, and 3 to specify the information to be carried in the automated data base.

5. Project the environment in which the methodology will operate. Come to terms with how the methodology will be hosted, accessed, and its use accounted for.

6. Choose a DBMS. Knowing the machine support available, the constraints, the information content of the data base, and the intended uses of the data; a management system (DBMS) for the automated requirements database may now be chosen.

7. Finally, and only now, define the language(s). Note that there may be very different language needs depending on the different aspects of communication, analysis and simulation. Even for the purpose of communication, levels of language may be needed according to the types of participants envisioned.

When AIRMICS has in this way defined the intent and nature of an Army information systems requirements methodology, the recommendations and evaluations of Section 4 and Appendix A will serve to establish a sufficient technology base for implementation. At that time it can be expected that existing technology will be suitable for transfer to the Army information systems environment.

# APPENDIX A ORGANIZATIONS SURVEYED

| | |
|---|---|
| A.1 | ARPA System Design Laboratory Project |
| A.2 | Ballistic Missile Defense Advanced Technology Center (BMDATC) |
| A.3 | Computer-Aided Design and Specification Tool (CADSAT) Project |
| A.4 | Computer Sciences Corporation |
| A.5 | Hughes Aircraft Company |
| A.6 | HOS, Inc. |
| A.7 | Integrated Computer-Aided Manufacturing (ICAM) Project |
| A.8 | Information Systems Design Optimization System (ISDOS) Project |
| A.9 | Martin Marietta Aerospace Company |
| A.10 | Naval Weapons Center, China Lake |
| A.11 | Softech, Inc. |
| A.12 | Sperry-Univac, Inc. |
| A.13 | System Development Corporation (SDC) |
| A.14 | Teledyne Brown Engineering, Inc. |
| A.15 | Xerox Corporation |

# METHODOLOGY DATA SUMMARY FORM

1. **Organization:** ARPA System Design Laboratory Project
   Naval Ocean Systems Center
   271 Catalina Blvd.
   San Diego, California 92152

2. **Methodology Involvement:**

   [ ] Developer [X] Evaluator/Integrator

   [ ] Sponsor [ ] User

3. **Availability:**

   [X] Public Domain [X] In-House [ ] Purchase or Lease

4. **Survey Vehicle:**

   [ ] Documentation [X] Personal Contact

5. **Potential User Community:**

   Navy Laboratories, embedded military computer systems developers.

6. **Methodology:**

   **Name(s):** System Design Laboratory Project (multiple methodologies)

   **Development Stage:** In partial use and evolving.

   **Activities Addressed (Figure 3.1):**
   All requirements phase activities.

   **Level of Support:**
   Automated tools provided on a distributed network with user documentation.

7. **Utility to AIRMICS:**

   [ ] Applicable now

   [X] Should be monitored <u>actively</u>

   [ ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

System Design Laboratory Project by Advanced Research Projects Agency (ARPA) and Naval Ocean Systems Center (NOSC.)

2. Sponsor Goals

To upgrade the technology available to designers and developers of embedded military computer systems.

3. Context

The System Design Laboratory Project is not itself a methodology for requirements definition. It is a project which is jointly sponsored by ARPA and the Navy to make newly emerging systems development technology readily available to embedded military computer systems developers. Along with the task of providing this technology, SDL also is responsible for evaluating, sponsoring, and integrating technology developments in areas where adequate tools and techniques are deemed to be lacking, such as is the case for the requirements specification and performance validation areas.

4. Approach to Activities Support

The SDL Project has involvement in all phases of system development from requirements specification through system testing. For example, in the requirements and design specification area the project has:

- sponsored development/evaluation of Higher Order Software methodology and its specification language (AXES).
- been instrumental in the initiation of the AN/UYK-7 version of URL/URA.
- conducted evaluative research in requirements/design methodologies and tools including URL/URA, HOS, and Parnas methodology.
- sponsored an ARPA-net installation of PSL/PSA.
- sponsored development of Hierarchical Design Methodology (HDM) and its associated tools.
- recently conducted a survey of tools and methodologies in use in Europe (report is forthcoming).

The above efforts were done in the project's role as integrator/evaluator/distributor of technology. Other work has centered on the development of a model SDL incorporating these results.

5. Dissemination/Use

The SDL model has been used experimentally by a few NOSC projects. Emphasis to date has been on validation of the SDL concept, rather than on production of an operational SDL.

6. Near-Term Direction

Attention will be turned toward producing an SDL which is operational and usable on real large-scale projects.

7. Utility

The work and charter of the SDL project is in many ways similar to that of AIRMICS. Many of the investigations, evaluations, and studies conducted and sponsored by the SDL project are of direct utility to AIRMICS.

8. Conclusions and Recommendations

It is recommended that AIRMICS establish liaison with the SDL project in order to share the results of the respective efforts of both organizations.

# METHODOLOGY DATA SUMMARY FORM

1. **Organization:** Ballistic Missile Defense Advanced Technology Center
Post Office Box 1500
Huntsville, Alabama 35807

2. **Methodology Involvement:**

   [ ] Developer  [X] Evaluator/Integrator

   [X] Sponsor  [ ] User

3. **Availability:**

   [X] Public Domain  [X] In-House  [ ] Purchase or Lease

4. **Survey Vehicle:**

   [X] Documentation  [X] Personal Contact

5. **Potential User Community:**
BMD and other real-time systems development communities.

6. **Methodology:**

   **Name(s):** Software Requirements Engineering Methodology (SREM)

   **Development Stage:** In use

   **Activities Addressed (Figure 3.1):**
   DP Requirements Structuring and Resource and Performance Projection Activities.

   **Level of Support:**
   Supported by an automated tool (RSL/REVS) on CDC 7600 and TI-ASC machines.

7. **Utility to AIRMICS:**

   [X] Applicable now

   [ ] Should be monitored

   [ ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

Software Requirements Engineering Methodology (SREM) by Ballistic Missile Defense Advanced Technology Center (BMDATC).

2. Sponsor Goals

To conduct an integrated software development research program to improve the techniques for developing correct, reliable BMD software.

3. Context

Bell, Bixler and Dyer summarize the context of SREM as follows:

> "SREM includes techniques and procedures for requirements decomposition and for managing the requirements development process. In addition SREM includes the Requirements Statement Language (RSL), a machine processible language for stating requirements, and the Requirements Engineering Validation System (REVS), an integrated set of tools to support the development of requirements in RSL."

4. Approach to Activities Support

The underlying concept of RSL/REVS is similar to that of PSL/PSA (see ISDOS write-up) in that it employs a machine processible language (RSL) as a medium of input to a relational data base called the Abstract System Semantic Model (ASSM). A set of automated tools (REVS) are used to analyze and report on the information in the ASSM. RSL/REVS, however, is more oriented toward the DP Requirements Structuring and Resource and Performance Projection Activities than PSL/PSA, as evidenced by its emphasis on flow-control and simulation. DP Requirements Structuring is supported via representations of processing hierarchy called R-Nets which consist of processing primitives called ALPHA's related via a small set of control-flow primitives. Simulations can be generated directly from the ASSM to support Resource and Performance Projection.

RSL/REVS is closely tied to the concepts of SREM. Even though the language is user-extensible, it is inherently less adaptable than PSL/PSA, because the core concepts of the language are given strict interpretation in SREM, whereas this is not the case for PSL. As is noted in the ISDOS discussion, this can be viewed as either beneficial or detrimental.

5. Dissemination/Use

SREM represents a major effort in methodology and tool development for requirements definition. It has gained recognition and application beyond the scope of BMD as evidenced by the following:

- Naval Air Development Center has an RSL/REVS installed on a CDC 6600.
- EPRI is using RSL/REVS as a tool for research in "ultra-reliable requirements definition" for reactor control.
- McDonnel Douglas Aircraft has used SREM experimentally.
- The Applied Physics Lab, John Hopkins University, is using SREM in a V&V role on a major switching system.

6. Near-Term Direction

Work is in progress to develop extensions to RSL/REVS to support distributed data processing.

7. Utility

RSL/REVS has been proven to be a valuable tool for requirements engineering. The extensibility of RSL makes it applicable in many environments, without loss of the core concepts of SREM. The development of distributed data processing features of the tool will be of great benefit to the requirements definition community since there does not seem to be adequate support for this aspect by other methodologies/tools. It should be noted that SREM is a large, integrated system requiring considerable machine resource (128K 60 bit words of memory).

The core constructs of RSL (R-nets, ALPHA's, etc.) are not inherently obvious as to their meaning. This requires a certain conceptual leap on the part of the users in order to relate RSL to their real-world applications, which in effect limits the applicability of the tool to the more conceptually oriented. Centralized support in the form of formal training courses and user group communication would help to alleviate this problem.

8. Conclusions and Recommendations

It is recommended that AIRMICS perform detailed investigation with respect to:

- Utility as a communications media among initiators and systems engineers during the Problem Concept Definition activity.
- The applicability of the level of the simulations.
- Resources required, e.g. computer, training, manpower.
- Transferability to the AIRMICS environment and potential problems of ongoing support.

# METHODOLOGY DATA SUMMARY FORM

1. **Organization:** Computer-Aided Design and Specification Tool (CADSAT) Project
   ESD/TOIT
   Hascom AFB
   Bethesda, Maryland 01731

2. **Methodology Involvement:**

   [ ] Developer  [X] Evaluator/Integrator

   [X] Sponsor  [ ] User

3. **Availability:**

   [X] Public Domain  [X] In-House  [ ] Purchase or Lease

4. **Survey Vehicle:**

   [X] Documentation  [X] Personal Contact

5. **Potential User Community:**
   Air Force: ESD, RADC; SAMSO; Navy: NUSC, FCDSSA

6. **Methodology:**

   **Name(s):** User Requirements Language/User Requirements Analyzer (URL/URA)

   **Development Stage:** Operational and in use

   **Activities Addressed (Figure 3.1):**
   Information Structuring and DP Requirements Structuring Activities.

   **Level of Support:**
   Automated support available on IBM, Honeywell and AN/UYK-7 machines with user documentation.

7. **Utility to AIRMICS:**

   [ ] Applicable now

   [X] Should be monitored

   [ ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

User Requirements Language/User Requirements Analyzer (URL/URA) by U.S. Air Force Electronic Systems Division (ESD).

2. Sponsor Goals

To provide automated tools to reduce cost and risk in Air Force system development projects.

3. Context

URL/URA is an Air Force sponsored version of PSL/PSA (see ISDOS Project write-up), produced under Air Force contract to the University of Michigan. Its development followed version 2 of PSL/PSA and included significant enhancements to the language and the analyzer. (Subsequent versions of PSL/PSA include most of the URA analyzer capabilities [PSA4.2] and language constructs [PSA5.1].)

4. Approach to Activities Support

ESD is in the role of providing automated tools to Air Force system development projects. The utility and flexibility of PSL/PSA was recognized and so the decision was made to make this tool available to ESD projects. After some initial use, it was determined that Version 2 of PSL/PSA was inadequate for military systems, and so ESD contracted with the University of Michigan to add language features (primarily system dynamics and traceability constructs) and analyzer facilities (added name selection and reporting capabilities).

The CADSAT Project's approach to supporting its users is essentially the same as ISDOS, which is to provide the tool (URL/URA) and user documentation, but not to impose or recommend methodology for its use. The determination of this methodology is up to the individual projects using URL/URA. One attempt was made to prepare a URL/URA guidebook, but project personnel do not feel that it was successful. ESD is expecting to gather documentation on URL/URA methodology being developed by contractors using the tool, and to disseminate this as examples of such guidelines.

5. Dissemination/Use

URL/URA has had use in several Air Force and Navy projects. These include:

- SEEKFROST
- Joint Surveillance System (JSS)
- E3A/AWACS System
- Integrated Computer-Aided Manufacturing (ICAM) Project

URL/URA is available on IBM 360/370 machines, Honeywell Multics, and the AN/UYK-7/SHARE-7 system at Air Force and Navy facilities.

6. Near-Term Direction

There are no plans at this time for conversion from URL/URA to PSL/PSA Version 5.1 by projects using the CADSAT system, even though the new PSL/PSA version encompasses most of the capability of URL/URA. At such time that the mainstream of PSL/PSA would represent a significant improvement over URL/URA, or maintenance of URL/URA becomes too costly, this conversion will be considered. As mentioned previously, CADSAT does not intend to enforce methodology for use of URL/URA, but rather to collect and share methodological guidelines developed by user projects, which is expected during the next six months to a year.

7. Utility

The utility of URL/URA is basically the same as PSL/PSA with a couple of exceptions:

- URL/URA is supported on a different set of host machines than PSL/PSA.
- URL/URA is out of the mainstream of PSL/PSA evolution and will not include some new PSL/PSA features such as graphics interface and query capability.

8. Conclusions and Recommendations

Because of the relationship between URL/URA and PSL/PSA Version 5.1, it is not necessary that AIRMICS closely monitor URL/URA. AIRMICS should, however, monitor URL/URA users as it would PSL/PSA users, and pay close attention to these projects' URL/URA methodology guidelines when they become available.

# METHODOLOGY DATA SUMMARY FORM

1. **Organization:** Computer Sciences Corporation
   6565 Arlington Blvd.
   Falls Church, Virginia 22046

2. **Methodology Involvement:**

   [ X ] Developer　　[ ] Evaluator/Integrator

   [ ] Sponsor　　[ X ] User

3. **Availability:**

   [ ] Public Domain　　[ X ] In-House　　[ ] Purchase or Lease

4. **Survey Vehicle:**

   [ X ] Documentation　　[ X ] Personal Contact

5. **Potential User Community:**
   Software systems development requiring extensive pre-delivery testing.

6. **Methodology:**

   **Name(s):** Digital Systems Development Methodology (DSDM); System Verification Diagrams (SVD); THREADS

   **Development Stage:** In use for certain in-house projects

   **Activities Addressed (Figure 3.1):** Problem Concept Definition; Information Structuring

   **Level of Support:** Engineering practices manuals

7. **Utility to AIRMICS:**

   [ ] Applicable now

   [ ] Should be monitored

   [ X ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. **Name and Sponsor**

Digital Systems Development Methodology (DSDM) by Computer Sciences Corporation.

2. **Sponsor Goals**

To develop a communicable methodology for software development across the entire development process. The primary goal is more cost effective software development as a creditable capability in the marketplace.

3. **Context**

The Computer Sciences Corporation is the largest hardware-independent software development house in the country. The organization is comprised of a number of semi-autonomous divisions operating from a large number of dispersed offices.

The methodology reflects the needs of the corporation to develop large systems in the context of a large, fluid workforce. The methodology therefore focuses on very fundamental issues which we summarize as:

- Communication between workers.
- Manageability of a development.

Manageability with respect to DSDM can be translated to mean the establishment and achievement of verifiable milestones along the development process. For each development phase, emphasis is on verification of the increments of work performed. In this context, the major methodology components are:

1. System Verification Diagram (SVD) techniques.
2. THREADS, a diagrammatic design verification technique.
3. BUILDS, an approach to bottom-up ongoing testing and integration.

4. **Approach to Activities Support**

The System Verification Diagram (SVD) technique is a diagrammatic problem definition language oriented toward the Information Structuring Activity.

An individual SVD box represents a stimulus-response requirement as determinable from the (prose) requirements definition. Inputs and outputs can be data and/or events. Procedures for establishing connectivity are provided. A connected chart of boxes is an SVD.

There is no generally available automated support within the organization for an SVD activity. Further, there is no established, practiced discipline for decomposition/structuring. The approach seems concerned only with path connectivity to the exclusion of the other requirements considerations enumerated throughout the study report. The purpose served is as the name indicates, verification of the requirements (document).

In any large undertaking, consistency and completeness of baseline requirements are of real value. SVDs would certainly be helpful with respect to connectivity concerns as a documentation tool. So too, and in much the same fashion, HIPO charts are of value. We can perceive no material advantage of employment of SVDs over a block diagram display such as HIPO charts.

Methods for the use of SVDs reflect the general concepts of good practices in the field of requirements engineering at large. Whatever their utility for other reasons, other requirements methodologies with diagrammatic input are available (Teledyne-Brown, Softech, Hughes, etc.), which seem more fully thought out.

Computer Sciences Corporation methodology has the distinctive trait of path linearization. That is, common portions of paths are not allowed to merge in the diagrammatic techniques. This is to preserve one-to-one stimulus-response testability. However, while this may be one valid motivation, major problems in use occur:

1. The documentation proliferates markedly.
2. Identification of commonalities, a key issue for design and implementation, are obscured.

5. Dissemination/Use

SVDs are a relatively new methodology component whose project use is not known to the authors. However, there is clear connection to THREADS in that SVDs appear to be THREADS reformulated from a design to a requirements context. Many of the same comments can be made for both. THREADS has been used on a number of projects which include: DDG47, the AEGIS ship; Carrier-based Tactical Support Center (CV-TSC); Ocean Surveillance Information System (OSIS); Goddard Real-Time System (GRTS).

6. Near-Term Direction

Disseminate software engineering handbooks within the organization.

7. Utility

The body of experience does not show the approach to be an acceptable documentation aid because of the difficulties in production and maintenance of the diagrams.

8. Conclusions and Recommendations

None

# METHODOLOGY DATA SUMMARY FORM

1. **Organization:** Hughes Aircraft Company
Design Analysis Section (606/K121)
Post Office Box 3310
1901 W. Malvern Avenue
Fullerton, California 92634

2. **Methodology Involvement:**

   [ X ] Developer  [ ] Evaluator/Integrator

   [ ] Sponsor  [ X ] User

3. **Availability:**

   [ ] Public Domain  [ X ] In-House  [ ] Purchase or Lease

4. **Survey Vehicle:**

   [ X ] Documentation  [ X ] Personal Contact

5. **Potential User Community:**
Experience is in embedded processing systems, but general purpose information systems are not precluded.

6. **Methodology:**

   **Name(s):** Design Analysis System (DAS)

   **Development Stage:** In use and evolving.

   **Activities Addressed (Figure 3.1):**
   Information Structuring, DP Requirements Structuring, and Resource and Performance Projection Activities.

   **Level of Support:**
   Software engineering standards manuals, user's manuals, interactive graphics, automated data base and analysis (PSL/PSA), and automated simulation.

7. **Utility to AIRMICS:**

   [X ] Applicable now

   [ ] Should be monitored

   [ ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

Design Analysis System (DAS) by Hughes.

2. Sponsor Goals

To develop comprehensive in-house software engineering support, corporate development standards, and software/systems quality assurance.

3. Context

Hughes has committed to Structured Design as its design methodology. The approach has until now been entirely manual, but considerable experience with Structured Design has been obtained.

Rather than simply develop automated support for the design process, Hughes has internally developed an integrated approach to requirements definition, simulation and design.

4. Approach to Activities Support

The Information Structuring Activity is supported by a functional flow diagramming technique: Information Flow Diagrams (IFDs). The DP Processing Requirements Structuring is supported by another diagramatic technique: Operational Flow Diagrams (OFDs). The OFD graphical language is supported by graphics terminals for input. Each problem being given an OFD description ends up represented in a PSL/PSA data base. The same approach is possible for Information Structuring by means of IFDs. However, this activity is manual and will likely remain so since in practice, there have been relatively few diagrams needed.

The IFD and OFD techniques have been kept purposely very simple for two reasons.

1. So that use is easy.
2. So that comprehension of diagrams is easy for the problem initiator/sponsor. The methodology calls for interaction between problem initiators and the requirements engineers.

In support of the validity of the requirements:

- PSA is invoked for analysis.
- Functional simulations can be generated automatically from the OFD PSA data base.

IFDs, OFDs, and the attendant graphics, PSL/PSA, and simulation tools should not be viewed outside of the context of the uniform approach which will provide the Structured Design Techniques with:

- Interactive graphics input.
- PSL/PSA data base.
- Distributed processing oriented automatic simulation.

5. Dissemination/Use

IFD and OFD techniques have just begun to have try-out experience. Interestingly, the techniques seem applicable to project management concerns.

The approaches have not been tried on data oriented general purpose information oriented problems. However, the authors feel that when complete, the total Hughes methodology will be of significant utility in information systems applications.

The methodology is intended for in-house use, but no proprietary posture is at present in evidence.

6. Near-Term Direction

The automation of Structured Design is projected to be operational in less than a year. All automation will be supported on an AMDAHL 470/V5 under OS/MVS/TSO.

Projects with which to showcase the methodology are being sought.

7. Utility

The authors perceive that the Hughes methodology will have a strong position in the embedded processing software/systems marketplace. The methodology is not specifically geared toward the traditional general purpose information system applications which concern AIRMICS. However:

1. Many aspects of information system requirements definition can be supported by the methodology.
2. Advanced systems, those with relationship to command and control needs, will require a methodology with certain of the capabilities emphasized by DAS.

8. Conclusions and Recommendations

It is recommended that AIRMICS:

1. Evaluate the projected baseline capabilities for the next year of internal Hughes development against AIRMICS requirements definition needs.
2. Investigate the design support capabilities of the methodology.

# METHODOLOGY DATA SUMMARY FORM

1. **Organization:** Higher Order Software, Inc.
806 Massachusetts Avenue
Cambridge, Massachusetts 02139

2. **Methodology Involvement:**

[ X ] Developer    [ ] Evaluator/Integrator

[ ] Sponsor    [ X ] User

3. **Availability:**

[ X ] Public Domain    [ X ] In-House    [ ] Purchase or Lease

4. **Survey Vehicle:**

[ X ] Documentation    [ X ] Personal Contact

5. **Potential User Community:** Systems with intricate control architecture requirements.

6. **Methodology:**

Name(s): HOS methodology which includes the AXES requirements specification language.

Development Stage: In use and evolving.

Activities Addressed (Figure 3.1): Information structuring

Level of Support: Papers in the open literature; trained in-house personnel.

7. **Utility to AIRMICS:**

[ ] Applicable now
[ ] Should be monitored
[ X ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

HOS by Higher Order Software, Inc.

2. Sponsor Goals

To provide a mathematical, formal approach for the specification of system requirements.

3. Context

HOS methodology has its origins in the need for reliable space mission software. Such software is characteristically complex in its real-time inter-relationships. The mathematical formality of the HOS approach can be traced to the understandable desire to produce provably reliable software systems. Toward this goal, HOS attempts to formalize the structural aspects of implementation independent requirements definitions.

4. Approach to Activities Support

The structuring methodology is emboddied in project reports and articles in the literature. AXES is a formal requirements specification language. A compiler for AXES is in progress, but particulars are not known to the authors.

The problem of precise mathematical formulation is a difficult one. There does not appear to be much available in the way of methods and tools to support a large requirements definition effort in a practical sense. In general, HOS approaches can be characterized as abstract.

5. Dissemination/Use

Higher Order Software, Inc. is a relatively new organization. HOS methodology has been applied in a few project situations, but the particulars are not known to the authors.

6. Near-Term Direction

Opportunity for the employment of AXES will be sought. Methodology R&D continues.

7. Utility

The abstractness of the HOS approach, and an apparent lack of fit to the Army context, contribute to the judgement that the methodology is of low utility to AIRMICS.

8. Conclusions and Recommendations

None

# METHODOLOGY DATA SUMMARY FORM

1. **Organization:** Integrated Computer-Aided Manufacturing (ICAM) Project
   Air Force Materials Laboratory
   Wright-Patterson AFB
   Dayton, Ohio 45433

2. **Methodology Involvement:**

   [ X ] Developer [ X] Evaluator/Integrator

   [ X ] Sponsor [ X] User

3. **Availability:**

   [X ] Public Domain [ ] In-House [ ] Purchase or Lease

4. **Survey Vehicle:**

   [ X ] Documentation [X ] Personal Contact

5. **Potential User Community:** Aerospace Computer Aided Manufacturing (CAM) in particular; CAM in general

6. **Methodology:**

   **Name(s):** IDEF Versions 0-2

   **Development Stage:** IDEF Verson 0 (Functional and Data Decomposition) defined; automating in progress. IDEF Versions 1 and 2 (Information Flow and System Dynamics respectively) under investigation.
   **Activities Addressed (Figure 3.1):** Information Structuring, Data Processing Requirements Structuring, and Resource and Performance Projection Activities

   **Level of Support:** SADT (softech), DAS (Hughes), ADBMS which is the ISDOS DBMS graphics terminals, CDC CYBERNET.

7. **Utility to AIRMICS:**

   [ ] Applicable now

   [ X] Should be monitored actively

   [ ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

Integrated Computer Aided Manufacturing (ICAM) project of the Air Force at the Air Force Materials Laboratory, Wright-Patterson Air Force Base.

2. Sponsor Goals

We quote objectives from the ICAM Program Prospectus:

> To perform manufacturing technology which will:
>
> -- Reduce defense systems costs by developing and applying computer-aided manufacturing technology to the fabrication of defense material.
>
> -- Establish a model for the integrated application of computer technology to all phases of production/manufacturing.
>
> -- Improve the long-term competence, efficiency and responsiveness of American aerospace and related industries to defense needs.
>
> -- Provide a mechanism for Integrated Computer-Aided Manufacturing technology transfer to and within American industry.
>
> -- Validate and demonstrate the cost-saving benefits and flexibility of ICAM for representative elements of Air Force Systems production.

3. Context

The ICAM project is one of great size and potential significance. To illustrate the level of effort, funding is approximately one million dollars through fiscal 1982.

The context of the project is made clear by the Abstract to the Program Prospectus which we reproduce here:

> The U.S. Air Force program for Integrated Computer-Aided Manufacturing (ICAM) was brought about by needs and pressures in state-of-the-art technologies, economics, increasing human limitations, aerospace design and manufacturing complexity, computer developments, and competition from abroad. These factors will bring ICAM of the American scene eventually, with or without an Air Force role. However, Air Force involvement is logical and desirable because the government is a large customer of manufacturing production, and because the ICAM program is a logical extension of

previous Air Force work in computer-aided manufacturing. As a primary goal, the ICAM program is a practical effort to greatly shorten the implementation timespan for incorporation of compatible and standardized techniques and to provide unified direction for industry. Of particular importance to the Air Force is surge production capability, improved cost-effectiveness of weapon system production, and flexible manufacturing capabilities required to maintain a credible defense posture. ICAM is essentially a program and development plan to produce systematically related modules for efficient manufacturing control. Modules may be separately developed, and may be individually implemented in industry, with short-term gains, but the primary benefits of the modular structure will be most evident in a fully-integrated system. The private sector is heavily involved in the program coordination, in which a "wedge" of sheet metal fabrication is being modeled and developed to demonstrate coordination by computer of phases of design and manufacturing. In addition to substantial cost savings and improved management control, ICAM will permit designs in which parts are computer-examined for performance evaluation and economical fabrication, and in which the computer will permit rapid examination of management choices in the detailed planning of manufacturing.

4. Approach to Activities Support

There is a fundamental understanding that the objectives of the project can be reached only through a comprehensive, appropriate methodology. Central to this methodology is the capability to baseline the present aerospace manufacturing processes, and demonstrate the effects of an integrated approach. This is expresssed succinctly in the Septermber 1978 quarterly project newsletter. The ICAM Project Report:

> As a starting point, ICAM manufacturing architecture must look at two major items: existing manufacturing architecture (as a combination of existing systems in the aerospace industry), and the integration goals of ICAM projects as they may impact the total scheme. The result is a model (actually a series of models at various levels) of logical functions needed to accomplish a manufacturing objective - much like an organization chart using production functions and interconnections rather than departments.

In a project devoted to developing computer aided manufacturing methodology, it is not surprising to see efforts in requirements methodology. ICAM is, at present, heavily involved in investigations across the spectrum of requirements activities. In particular:

1. IDEF Version 0 - Functional and data decomposition using the SADT approach of Softech. Baselines were produced manually. Automation is in progress.

2. IDEF Version 1 - Information Flow using the Hughes DAS system accessed via the CYBERNET.
3. IDEF Version 2 - System dynamics and the operational view. No one approach is as yet distinguished; a number of modeling techniques are under investigation.

It is intended that a fully automated system definition capability be developed which will encompass:

- Graphics terminal input.
- Remote access supported by the CYBERNET.
- Many levels of static and dynamic models obtainable in an interactive user oriented environment.
- Machine archiving.

5. Dissemination/Use

The ICAM project is to be a user of its own methodology. Considerable hands-on trial of candidate methodology components is intended.

The project maintains a high level of visibility; effort is made to disseminate results and experience.

6. Near-Term Direction

Automation of the functional and data decompositions of IDEF Version 0 is being pursued. The DBMS will be ADBMS of ISDOS.

Specific models and a user (language) interface will be developed.

7. Utility

The context of ICAM is markedly different from that of AIRMICS. It is felt however, that technical accomplishments and lessons learned by ICAM could be very valuable to AIRMICS.

8. Conclusions and Recommendations

It is strongly recommended that active liaison with ICAM be established.

# METHODOLOGY DATA SUMMARY FORM

1. Organization: Information Systems Design Optimization System (ISDOS) Project
   University of Michigan
   Ann Arbor, Michigan 48109

2. Methodology Involvement:

   [ X ] Developer     [ ] Evaluator/Integrator

   [ ] Sponsor     [ ] User

3. Availability:

   [ X ] Public Domain     [ ] In-House     [ ] Purchase or Lease

4. Survey Vehicle:

   [ X ] Documentation     [ X ] Personal Contact

5. Potential User Community:
   In use in a variety of systems spanning military and commercial environments.

6. Methodology:

   Name(s): Problem Statement Language/Problem Statement Analyzer (PSL/PSA).

   Development Stage: In use (fifth major revision)

   Activities Addressed (Figure 3.1):
   Information Structuring and Data Processing Requirements Structuring Activities.

   Level of Support:
   Automated analyzer on many host machines, large amounts of documentation, yearly user workshops, project review meetings, and training classes.

7. Utility to AIRMICS:

   [X ] Applicable now

   [ ] Should be monitored

   [ ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

Problem Statement Language/Problem Statement Analyzer (PSL/PSA) by ISDOS Project, University of Michigan.

2. Sponsor Goals

The ISDOS project is supported by a large number of sponsors. The long-term goal of the project is to improve the system development process by providing (automated) tools to aid in the activities of system development.

3. Context

The first step toward the above goal has been the development of a computer-aided tool for requirements documentation called PSL/PSA. PSL/PSA is not itself a methodology for requirements definition. It is a documentation tool which imposes structured, formal thinking but lacks formal methodology for its use. Users of PSL/PSA must either implicitly or explicitly define this methodology.

4. Approach to Activities Support

ISDOS does not have an approach to activities support, other than to suggest the use of PSL/PSA. The tool has been supported by ISDOS for the Information Structuring and DP Requirements Structuring Activities. Application of PSL/PSA to the Resource and Performance Projection Activity has been done by others (Hughes, Martin Marietta) with little support from ISDOS. Earlier versions of PSL/PSA were more closely aligned with Information Structuring, than with DP Requirements Structuring, but later versions appear to be equally applicable to both activities, due to the addition of system dynamics description facilities.

The basic concept of PSL/PSA is that of a formal computer-processible language (PSL) which is used it construct a computer stored relational data base which may be updated incrementally. Documentation of the system description thus stored is cbtained by generating reports using PSA. While this basic concept has not changed, the details of the language and analyzer have evolved as the result of user impact at annual user workshops and project review meetings.

PSL/PSA is highly flexible. It is this flexibility which allows for the variety of applications and creates the need for methodology for its use in the requirements definition process. This can be both a detriment or a benefit. Lack of a prescribed formal methodology can result in ineffective or counterproductive use of the tool, but also makes the tool easily adaptable to a wide variety of situations.

PSA is supported on a variety of host machines (IBM, Honeywell, DEC, UNIVAC). ISDOS provides user documentation, software support, training and user group communication to the project sponsors.

5. **Dissemination/Use**

PSL/PSA has had widespread dissemination and use in both the commercial and military communities. Applications have varied in activity from high level process structuring to preliminary design, and in level of use from small experimental applications to full-scale uses including major modifications to the language and/or analyzer. Organizations (projects) which have used PSL/PSA include:

- U.S. Air Force (Joint Surveillance System, JSS, Grand-Based Electronic-Optical Deep Space Surveillance System CEDDS).
- U.S. Navy (DDG-47 Combat System Architecture, Restructured NTDS, a Fleet Materiel Support Office project).
- Chase Manhattan Bank (eight major projects)
- Hughes Aircraft Company
- Southern California Edison
- Univac - Roseville
- AT&T Long Lines (41 projects)
- Martin Marietta (Space Shuttle Altitude Control System)

## 6. Near-Term Direction

ISDOS is currently performing acceptance testing of PSL/PSA Version 5.1. This version incorporates all language features of past PSL versions as well as adding some new features. A number of existing reports have been enhanced and new reports added. An interface to plotter graphics for generation of structure and flow diagrams has been constructed. Plans for Version 5.2 include a query system and automated completeness specification/verification facility.

ISDOS has no plans for developing their own PSL/PSA "methodology", although the project has succumbed to user pressure to write a "PSL/PSA Applications Guidebook". In recognition that even more flexibility is necessary for use of PSL/PSA with a particular methodology in a particular environment, ISDOS has built a PSL/PSA meta-system. This meta-system allows automatic generation of different PSL/PSA systems based on a user-supplied description of the particular PSL desired. The meta-system has been used to construct a PSL/PSA system based on the NWC methodology (see appendix A.10). This development is consistent with the observations of Section 4.6, which discusses the importance of the DBMS and methodology to requirements definition.

## 7. Utility

The current revision of PSL/PSA seems to reflect the requests and needs of the user community without the need for any major change to the basic PSL/PSA concept. The implementation of PSA (based on ADBMS, written mostly in ANSI FORTRAN IV) is especially adaptable to varied environments. The combination of a sound basic concept, easily adaptable software, and lack of rigid methodology makes its utility as a requirements documentation and analysis support tool extremely high.

8. Conclusions and Recommendations

It is recommended that AIRMICS:

- Monitor the developments of PSL/PSA related tools and methodologies.
- Evaluate PSL/PSA's applicability to the AIRMICS environment with focus on:
  - methodology for PSL/PSA use
  - use of PSL/PSA with existing methodologies
  - existing adaptations of PSL/PSA (CADSAT, RDL/RDP, MEDL-R, etc.)

# METHODOLOGY DATA SUMMARY FORM

1. **Organization:** Martin Marietta Aerospace Company
   Post Office Box 179
   Denver, Colorado 80201

2. **Methodology Involvement:**

   [ X] Developer  [ ] Evaluator/Integrator

   [ ] Sponsor  [X ] User

3. **Availability:**

   [ ] Public Domain  [X ] In-House  [ ] Purchase or Lease

4. **Survey Vehicle:**

   [ X] Documentation  [X ] Personal Contact

5. **Potential User Community:**
   Experience is in real-time systems.

6. **Methodology:**

   **Name(s):** No single generic name.

   **Development Stage:** In use and evolving.

   **Activities Addressed (Figure 3.1):**
   DP Requirements Structuring and Resource and Performance Projection Activities.

   **Level of Support:** Machine-readable structuring languages with automated data base management; automated graphical analysis, automated simulation; user's manuals; software engineering standards manuals.

7. **Utility to AIRMICS:**

   [ ] Applicable now

   [ X ] Should be monitored

   [ ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

Martin Marietta Aerospace

2. Sponsor Goals

To develop a comprehensive in-house software development capability.

3. Context

The organization has extensive software/systems experience in the embedded processing area. The set of software engineering standards manuals document the corporate approach and establish development guidelines over the whole of the life cycle.

The approach to tools automation, and methodology in general, reflects clear in-house purposes. Short learning time, ease of use, and bounded, obtainable objectives are considered the important goals for the methodology. The result is a no frills set of core capabilities with which to support the engineering efforts.

4. Approach to Activities Support

Martin Marietta has been a direct sponsor of ISDOS, and has had considerable involvement with PSL/PSA. A current project, independent V&V for the flight and ground control software for the Air Force Shuttle, makes use of PSL/PSA. An automated transactional simulation generation tool has just recently been developed. The generator accesses the PSA data base which is enhanced by Comment Entries for purposes of simulation dynamics.

Another tool which operates off of the PSA data base is GAPS, a graphical analysis tool. GAPS is used to:

- Diagram the PSL relationships.
- Display clustering and architectural aspects for purposes of engineering the structuring.

An IR&D effort has recently produced a requirements language, MEDL-R. MEDL-R is supported by a relational DBMS with CRT interactive input on a

PDP 11/70. The language is based on PSL, but has been modified and streamlined. MEDL-R is a smaller capability than PSL whose purpose is simplicity in training and production use. The productivity gained is at the expense of generality. The approach represents what is possible when there is a body of experience applicable to a specific class of problems.

5. Dissemination/Use

There appears to be no direct intention to export methodology; conversely, it seems doubtful that tools would be held proprietary.

Internally, MEDL-R has not yet been used in project support.

6. Near-Term Direction

It is intended that MEDL-R be exercised. MEDL-D is a companion design language under development. It is intended that MEDL-D be completed and also exercised.

There does not seem to be interest in diagramatic requirements input.

The methodology has not been applied to any general purpose information system application.

7. Utility

The methodology is not directly applicable nor directly transferable to AIRMICS. Naturally, the use of PSL/PSA is an exception.

The purposes of the diagramatic analysis tool, GAPS, are too specific for AIRMICS needs. Also, it is felt that diagramatic analysis should be in the form of analytical support to an automated diagramatic requirements language.

8. Conclusions and Recommendations

Knowledge of the techniques employed would be useful to AIRMICS, and further efforts should be monitored.

# METHODOLOGY DATA SUMMARY FORM

1. **Organization:** Electronic Warfare Test and Evaluation System (ECHO Range) at the Naval Weapons Center (NWC), China Lake, California

2. **Methodology Involvement:**

   [ ] Developer                [ ] Evaluator/Integrator

   [ ] Sponsor                [X] User

3. **Availability:**

   [X] Public Domain    [ ] In-House    [ ] Purchase or Lease

4. **Survey Vehicle:**

   [X] Documentation                [X] Personal Contact

5. **Potential User Community:**

   Real-time and distributed applications

6. **Methodology:**

   Name(s): ECHO Range Methodology

   Development Stage: In use

   Activities Addressed (Figure 3.1): Information Structuring and DP Requirements Structuring

   Level of Support: Univac 1108 and 1110 automation; structuring technique and PSL/PSA context manual

7. **Utility to AIRMICS:**

   [ ] Applicable now

   [X] Should be monitored

   [ ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

ECHO Range Methodology at the Naval Wespons Center (NWC), China Lake.

2. Sponsor Goals

The Electronic Warfare Test and Evaluation System (ECHO Range) is in the process of range upgrade. Additional range capabilities will include real-time range control. ECHO Range staff employ methodology for purposes of: requirements structuring, design structuring, documentation, and communication.

3. Context

Software development and maintenance is provided to the range by a combination of contractor and staff personnel. In addition to the range upgrade program, the need for new tests sets ongoing requirements for timely software modifications and additions.

Software engineering methodology was installed at the range by the authors of this report so as to provide:

1. Technical leverage for the software personnel so that modification schedules could be met.
2. Control over the (sometimes fractionated) software efforts.

4. Approach to Activities Support

A real-time oriented structuring methodology was given a PSL/PSA context for machine support. The methodology provides process flow primitives, and establishes design responsibilities for inter-task communication in a distributed environment.

The machine supported methodology runs on the ECHO Range Univac 1108, and on the NWC Central Site Univac 1110.

5. Dissemination/Use

The structuring methodology has been used for:

- Design verification in SURTASS.

- System requirements structuring for B-52 avionics update.
- Data processing requirements study for the Air Force All Weather Tactical System (AWTS).
- Air Force re-configurable missile data processing system requirements studies.
- Air Force High Energy Laser fire control design.

The ISDOS project has used the ECHO Range structuring methodology as a means to demonstrate the capabilities of the PSL/PSA metalanguage to be provided with the new release, 5.1.

6. Near-Term Direction

The authors expect to add to the methodology as new needs arise. In particular, increased information structuring capability and structuring of operator interactions are near-term areas of intended extension.

In can be anticipated that the methodology will find other users at NWC since it is supported at NWC's computation center.

7. Utility

The utility of the methodology for AIRMICS is, at present, low for two reasons:

1. The methodology is heavily process flow oriented.
2. Documentation for the methodology is highly technical for purposes of real-time structuring. Details on use of the approach in a project context are not included.

However, aspects of the methodology related to distributed system issues would be of use to AIRMICS.

8. Conclusions and Recommendations

It is recommended that progress in the methodology be monitored for development of techniques in areas of importance to information systems.

# METHODOLOGY DATA SUMMARY FORM

1. Organization: Softech, Inc.
   460 Totten Pond Road
   Waltham, Massachusetts 02154

2. Methodology Involvement:

   [X] Developer　　[ ] Evaluator/Integrator

   [ ] Sponsor　　[X] User

3. Availability:

   [ ] Public Domain　　[X] In-House　　[X] Purchase or Lease

4. Survey Vehicle:

   [X] Documentation　　[X] Personal Contact

5. Potential User Community:
   Primarily, information systems applications

6. Methodology:

   Name(s): Structured Analysis and Design Technique (SADT)

   Development Stage: Developed and in use

   Activities Addressed (Figure 3.1): Problem Concept Definition, Information Structuring

   Level of Support:
   Manuals, courses; machine support via PSL/PSA on two projects

7. Utility to AIRMICS:

   [ ] Applicable now
   [X] Should be monitored
   [ ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

Software Analysis and Design Techniques (SADT) by Softech, Inc.

2. Sponsor Goals

To provide a comprehensive methodology to support the requirements phase of software development.

3. Context

The methodology has evolved into an extensive problem definition discipline. Experience has been primarily with information systems in a commercial environment. More recent efforts have included a variety of system types.

SADT classes are a Softech product. As such, the methodology itself has been clouded by the proprietary interests of the company. There is some core of feeling among the methodologists contacted during the study that SADT is more elaborate and complex than is appropriate.

It should be noted that in terms of methods and procedures, this methodology represents, perhaps, the most fully developed treatment of its kind to be found.

4. Approach to Activities Support

SADT takes up the people aspect of getting a problem defined. It is intended that the methods and practices provide for communication and documentation. Different views of the same problem situation are encouraged.

Diagrammatic decomposition techniques support representations of the different views of the system. These representations help to bring out all of the issues of the problem. The documentation techniques have traditionally been manual. It is currently clear that manual approaches alone are inadequate to the needs of decomposition verification and ongoing maintenance for large problems. This would especially be the case considering that the SADT approach produces various problem views. For problems of large size, automated enforcement of consistency between the decompositions representing these views would be necessary.

The SADT approach is quite formal, but the methodology is not oriented to technically proficient information systems professionals. It is perhaps for this reason that the approach can seem over done to technical professionals. Technical professionals tend to drive toward a core of problem requirements such as to bound the system requirements. Then, they iterate between the informational definition of the problem and the data processing system requirements definition activities as shown in Figures 3.1 through 3.3.

The SADT methodology is not oriented toward the technical issues of data processing systems and is not supportive of the needs of the Data Processing Requirements Structuring Activity nor of the Design Activity as set forth by the Requirements Activities Models of Section 3.

5. Dissemination/Use

SADT has been used in a considerable number of circumstances. Examples of current employment of the methodology are:

- A financial management system for a government agency. PSL/PSA is being used for machine support.
- Army training system modeling for purposes of training modification evaluation.
- Functional decomposition for the Air Force Integrated Computer Aided Manufacturing (ICAM) project. (The ICAM project, and its use of SADT, is reported as Appendix A.7).

6. Near-Term Direction

Softech recognizes the importance of machine support, and would intend to find opportunity to establish connection with PSL/PSA.

7. Utility

There can be no doubt that certain of the methodological techniques in problem definition would be valuable for AIRMICS to incorporate. However, it would take some effort on the part of AIRMICS to determine those specifically applicable aspects of SADT and place them in an Army information system context.

## 8. Conclusions and Recommendations

It is recommended that AIRMICS monitor SADT capabilities and automation efforts, particularly as developed by the ICAM project.

# METHODOLOGY DATA SUMMARY FORM

1. **Organization:** Sperry Univac, Inc.
   Post Office Box 3942
   2276 Highcrest Drive
   Roseville, Minnesota 55113

2. **Methodology Involvement:**

   [X] Developer [ ] Evaluator/Integrator

   [ ] Sponsor [X] User

3. **Availability:**

   [ ] Public Domain [X] In-House [ ] Purchase or Lease

4. **Survey Vehicle:**

   [X] Documentation [X] Personal Contact

5. **Potential User Community:**
   In-house projects.

6. **Methodology:**

   **Name(s):** Requirements and Development Language/Requirements and Development Processor (RDL/RDP)

   **Development Stage:** Initial Use

   **Activities Addressed (Figure 3.1):**
   Information Structuring, DP Requirements Structuring, Resource and Performance Projection, Preliminary Design.

   **Level of Support:**
   RDP available on UNIVAC 1110, user's reference manual available.

7. **Utility to AIRMICS:**

   [ ] Applicable now

   [X] Should be monitored

   [ ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

Requirements and Development Language/Requirements and Development Analyzer (RDL/RDP) by Sperry-Univac, Inc.

2. Sponsor Goals

To provide a single documentation medium/tool to satisfy documentation requirements in all aspects of software development.

3. Context

RDL/RDP is based upon the ISDOS PSL/PSA. The intention, however, is for RDL/RDP to provide for a single-source description of all aspects of software development, including project management, requirements definition, design, implementation, testing, and performance analysis. It has been developed as an internal tool for use on in-house projects.

4. Approach to Activities Support

The PSL/PSA concept of formal language/data base/ analyzer is central to RDL/RDP. RDL is a much more complex extension of PSL with facilities addressing each of the software development aspects mentioned above. Neither the tool nor methodology for its use is yet fully developed, and so the specifics of the approach to supporting each activity of software development (not to mention requirements definition) is yet to be defined.

5. Dissemination/Use

RDL/RDP is at present being used experimentally on in-house projects. It is not yet available outside Univac.

6. Near-Term Direction

Plans are to validate the RDL/RDP concept on in-house pilot projects and, assuming successful results, make the tool available for use on customer projects and eventually become a Univac standard.

7. Utility

In its current stage of development, RDL/RDP is not of direct utility to AIRMICS. It does show, however, what sort of things can be done using PSL/PSA as a basis.

8. Conclusions and Recommendations

While RDL/RDP does attempt to address a much larger area than requirements definition, those parts which do address this area are of enough interest to justify monitoring by AIRMICS. Of equal, if not greater interest, is the determination of whether the underlying concepts of PSL/PSA can be successfully extended to encompass the entire realm of software development. It is therefore recommended that AIRMICS monitor the progress of RDL/RDP.

# METHODOLOGY DATA SUMMARY FORM

1. Organization: System Development Corporation
   2500 Colorado Avenue
   Santa Monica, California 90406

2. Methodology Involvement:

   [X] Developer   [ ] Evaluator/Integrator

   [ ] Sponsor   [X ] User

3. Availability:

   [ ] Public Domain   [X] In-House   [ ] Purchase or Lease

4. Survey Vehicle:

   [ ] Documentation   [X ] Personal Contact

5. Potential User Community:
   Experience primarily with weapon systems; certain aspects have potential for wide range of applications.

6. Methodology:

   Name(s): Software Factory

   Development Stage: In use and evolving.

   Activities Addressed (Figure 3.1):
   All requirements phase activities.

   Level of Support:
   Procedures manuals; manual diagramming.

7. Utility to AIRMICS:

   [ ] Applicable now
   [X ] Should be monitored
   [ ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

Software Factory by SDC

2. Sponsor Goals

To reduce development problems for contracted systems through greater attention to the requirements phase.

3. Context

Emphasis is on fundamentals: methods and techniques for taking care to do the requirements job well.

The methodology is developed and used by the Software Requirements and Analysis (SWRA) Department. The SWRA project support responsibilities extends actively through the design phase, and throughout the development in a requirements maintenance role. Therefore, there is attention to the designability and testability of the requirements produced.

4. Approach to Activities Support

The SWRA approach emphasizes information structuring prior to functional flow analysis. As further intent to define the problem, the methodology calls for development of a users handbook as early as possible. We quote from an SWRA internal working documents:

> Many experts in the systems analysis field espouse the 'user's manual first' theory. Essentially, the concept states that detailed functional analysis cannot proceed effectively unless the purpose or use of the data system is understood. Otherwise you end up with a solution in search of a problem. The operational concept or mission description must describe the role of the data system. --- Finally, an obsession with functionality at the expense of the data viewpoint has heavily contributed to system failures in the past.

The user's manual here provides a relatively formal context for Problem Concept Definition. A diagramming technique, Access Graphs, supports the Information Structuring Activity. HIPO charts have been employed in support of Data Processing Requirements Structuring. Experience rather than developed tools seems to be at the core of the Resource and Performance Projection Activity. However, simulation development is supported by an SDC developed language, MODLIT. MODLIT is an interactive dialect of GPSS.

5. Dissemination/Use

The Software Factory Methodology has had application to various in-house projects. Not all aspects of the methodology have been applied in each instance, but a body of corporate experience in use of its approaches to requirements definition exists. Relevant in-house projects include:

- Moroccan Air Defense System (MADS)
- Air Traffic Control for the German Defense Ministry (TRACKER)
- Mobile Sea Range (MSR), a real-time shipboard data acquisition system for the Navy.
- Operational Analysis Special Intelligence System (OASIS)
- Emergency Command and Control System (ECCS) for the Los Angeles Police Department

6. Near-Term Direction

The SWRA Department will be actively engaged in:

- Examining other diagramatic techniques; specifically SADT by Softech.
- Giving automated support to their techniques; specifically by use of PSL/PSA.

7. Utility

There are no specific, transferable tools available at this time, The concepts and methods could be quite applicable. Near-term developments (9 months) will produce relevant methodology.

8. Conclusions and Recommendation

It is recommended that AIRMICS:

1. Become familiar with the Software Factory concepts and methods.
2. Continue to monitor automation developments.

# METHODOLOGY DATA SUMMARY FORM

1. Organization: Teledyne Brown Engineering, Inc.
   300 Sparkman Drive
   Huntsville, Alabama 35807

2. Methodology Involvement:

   [ X] Developer [ ] Evaluator/Integrator

   [ ] Sponsor [ X ] User

3. Availability:

   [ ] Public Domain [ X ] In-House [ ] Purchase or Lease

4. Survey Vehicle:

   [ X] Documentation [ X ] Personal Contact

5. Potential User Community:
   Any development employing HIPO Charts could benefit by using this approach instead.

6. Methodology:

   Name(s): Input/Output Requirements Language (IORL)

   Development Stage: In use and evolving

   Activities Addressed (Figure 3.1):
   Information Structuring and Data Processing Requirements Structuring.

   Level of Support:

   Development control procedures and (Diagramatic) Language User's Manuals; interactive graphics.

7. Utility to AIRMICS:

   [ X ] Applicable now
   [ ] Should be monitored
   [ ] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

Input/Output Requirements Language (IORL) by Teledyne Brown Engineering (TBE).

2. Sponsor Goals

To enhance system development by relieving the well known problems of a good requirements definition.

3. Context

IORL encompasses a set of tabular parameter documentation methods and functional flow diagramming techniques. It is intended that the approach provide a consistent, evolving documentation of the development. It is also intended that the methodology both encourage and provide a natural environment for good practices.

4. Approach to Activities Support

IORL is an example of a formalization of internal system engineering procedure related primarily to weapon systems. As such, and for the reasons discussed in Section 3.1.3, we find IORL much more aligned with DP Requirements Structuring and Design than with Information Structuring.

The approach strongly, and appropriately, focuses on the data interfaces of functional modules. The approach also provides documentation context for functional decomposition. Decomposition here simply refers to functional and data detailing.

IORL is basically a graphical language with supporting tabulation. Further, it is functional flow oriented. As seems to be generally the case for approaches of this kind, system control and other system architectural concerns are not addressed with a hierarchical methodology. However, it must be pointed out that the diagramitic techniques include timing, resource and constraint considerations. This is in the form of diagramatic elaborations (IORTDs) of the function flow representations (SBDs).

The diagramatic techniques are fully supported by an interactive graphics terminal. The terminal is a 17 inch display with lightpen. The system is hosted by a PDP 11/34.

5. Dissemination/Use

IORL had its origins at Bell Telephone Laboratories and had been used on a portion of Safeguard. At that time it was also used for an MIS, the Investment and Cost Information System (ICIS).

Currently, at TBE, IORL is in use for V&V purposes on AN/TSQ() and Patriot projects. IORL has also found use in TBE proposals for the Battery Computer System and the Harrasement Weapon System.

6. Near-Term Direction

Automated Simulation and analysis is intended for the near future.

7. Utility

The diagramatic techniques together with their mini-computer hosted graphics support are somewhat unique in the field. Further, the approach is clearly along lines which would support the AIRMICS problem area.

The capability as is, stands alone as a front-end requirements documentation tool. In this sense, IORL is applicable to the AIRMICS problem now. However, it is also that case that in its present form, IORL does not provide a comprehensive methodology.

8. Conclusions and Recommendations

The requirements representation capability of the IORL system should be evaluated as to its ability to fulfill the requirements input language needs of AIRMICS.

# METHODOLOGY DATA SUMMARY FORM

1. Organization: Xerox Corporation
   Xerox Square 052
   Rochester, New York 14644

2. Methodology Involvement:

   [ X] Developer    [ ] Evaluator/Integrator

   [ ] Sponsor    [X ] User

3. Availability:

   [ ] Public Domain    [X ] In-House    [ ] Purchase or Lease

4. Survey Vehicle:

   [ X] Documentation    [X ] Personal Contact

5. Potential User Community:
   In-house application to business information systems.

6. Methodology:

   Name(s): Xerox System Methodology (XSM)

   Development Stage: Under development

   Activities Addressed (Figure 3.1):
   Information Structuring and DP Requirements Structuring Activities.

   Level of Support:
   Procedure Manual and an automated data dictionary.

7. Utility to AIRMICS:

   [ ] Applicable now
   [ ] Should be monitored
   [ X] Marginally useful or not applicable.

# METHODOLOGY REPORT

1. Name and Sponsor

Xerox Systems Methodology (XSM)

2. Sponsor Goals

To develop an internal methodology which address each phase of the life cycle.

3. Context

Xerox undertakes small, large, and even multi-national business information system developments. An extensive set of both technical and management guides in the form of manuals exists. These manuals serve as corporate standards. XSM has as purpose support of the corporate standards through the provision of techniques and tools.

4. *Approach to Activities Support*

Many of the system developments have been able to assume:

- A main-frame computer
- A commercially available data base management system (DBMS)

With the system technology prescribed, the Problem Concept Structuring Activity is accomplished by:

1. Defining and structuring the data base.
2. Defining output reports in terms of DBMS operations.

In this way, the Problem Concept Structuring and the DP Requirements Structuring Activities merge.

An automated data dictionary tool supports data base definition and structuring. The automation comes from an IBM 360 hosted software package called Datamanger and developed by Management System and Programming of London.

Diagramatic techniques are not currently incorporated within XSM. This is not to suggest that such techniques are disallowed. Rather, a standard is not yet determined.

XSM has not formulated a strong approach to the Resource and Performance Projection Activity.

PSL/PSA has been used by Xerox in a manufacturing process documentation context.

5. Dissemination

XSM is, at present, in a try-out mode of employment by several in-house projects.

6. Near-Term Direction

Graphical techniques will be evaluated. SADT is deemed as overly complex and will not be used.

The utility of PSL/PSA will be investigated.

7. Utility

The methodology is judged as not yet meaningfully transportable outside of the Xerox environment context.

8. Conclusions and Recommendations

The problems addressed by Xerox are relevant to the areas of CSC responsibility, although tools and transferable development methodology may come late for AIRMICS purposes. But AIRMICS should not overlook the degree of corporate experience in managing information system development. It is therefore recommended that at the appropriate point in the AIRMICS program, the Xerox project management approach be reviewed.

# APPENDIX B

# BIBLIOGRAPHY

AD-A075 227 BATTELLE COLUMBUS LABS DURHAM N C F/G 5/2
A SURVEY OF THE STATE-OF-THE-FIELD OF REQUIREMENTS METHODOLOGY.(U)
MAY 79 D B DEVORKIN, R T OBERNDOR DAAG29-76-D-0100

UNCLASSIFIED NL

2 OF 2

AD
A075227

END
DATE
FILMED
11-79
DDC

## Bibliography


Asch, A.; Connors, T.; Kelliher, D.W.; and Locher, J.P. III. DOD Weapon Systems Software Acquisition and Management Study. Volume 1. MITRE Findings and Recommendations, May 1975.

Alford, Mack W. "A Requirements Engineering Methodology for Real-Time Processing Requirements." IEEE Transactions on Software Engineering, Volume SE-3, No. 1 January 1977.

Alford, M.W. "Software Requirements Engineering Methodology (SREM) at the Age of Two", Proceedings of COMPSAC 78, IEEE Catalog No. 78CH1338-3C, November 1978.

Belford, P.C.; Bond, A.F.; Henderson, D.G.; and Sellers. L.S. "Specifications, A Key to Effective Software Development." Proceedings of Second International Conference Software Engineering, October 1976.

Bell, Thomas E.; Bixler, David C.; and Dyer, Margaret. "An Extendable Approach to Computer-Aided Software Requirements Engineering." IEEE Transactions on Software Engineering, Volume SE-3, No. 1 January 1977.

Bianchi, M.H., and Wood J.L. "A User's Viewpoint on the Programmer's Workbench." Proceedings of Second International Conference on Software Engineering, 1976.

Boyd, Donald L., and Pizzarello, Antonio. "Introduction to the Wellmade Design Methodology." IEEE Transactions on Software Engineering, Volume, SE-4. July 1978.

Brackett, John W., and Irvine, C.A. "Automated Software Engineering Through Structured Data Management." IEEE Transactions on Software Engineering, Volume SE-3, No. 1. January 1977.

COMPSAC 78, Computer Software Applications Conference, proceedings of, IEEE Catalog No. 78CH1338-3C. November 13-16, 1978.

Corrigan, Michael J., and Elston, Jay P. Design Document for the Functional Simulation Generator. Martin Marrietta, Inc., Denver, Colorado, 1979.

Couger, J.D. "Evolution of Business System Analysis Techniques." ACM Computing Surveys, Volume 5, No. 3. September 1973.

Cushing, Steven. "Algebraic Specification of Data Types in Higher Order Software (HOS)". Proceedings of Eleventh Hawaii International Conference on Systems Sciences, Honolulu, Hawaii, January 5-6, 1978.

DOD Weapon Systems Software Management Study. Applied Physics Laboratory. Johns Hopkins University. June 1975

Davis, Carl G., and Vick, Charles R. "The Software Development System." IEEE Transactions on Software Engineering, Volume SE-3, No. 1. January 1977.

DeLavigne, Kenneth T. "Basic Program Design - The Jackson Way: An Example." IBM Technical Report. TR 00.2762. June 3, 1976.

Densmore, Dana. LANDSAT NAVPAK System Design Analysis as a Demonstration of CSDL Software Engineering Technology. Charles Stark Draper Laboratory, Inc., Cambridge, Mass. April 30, 1977.

DeVorkin, Donald, and DeWolf, J. Barton. Critical Issues in Data Processing Requirements Specification. Charles Stark Draper Laboratory, Inc., Cambridge, Mass. May 17, 1977.

DeVorkin, Donald B. Echo Range Software Methodology Report. Computer Sciences Corporation, San Diego, California. July 17, 1978.

DeWolf, J. Barton, and Principato, Robert N. A Methodology for Requirements Specification and Preliminary Design of Real-Time Systems. Charles Stark Draper Laboratory, Inc., Cambridge, Mass. July 1977.

DeWolf, J. Barton; Rye, Phyllis; and Whitworth, Mark. Requirements Engineering -- An Approach to Formalization, Integration and Evaluation. Charles Stark Draper Laboratory, Inc., Cambridge, Mass. October 1978.

Dolotta, T.A., and Mashey J.R. "An Introduction to the Programmer's Workbench." Proceedings of Second International Conference on Software Engineering, 1976

Dyer, M.E. REVS Users Manual. Ballistic Missile Defense Advanced Technology Center, Huntsville, Alabama, August 17, 1977.

Dzida, W.; Herda, S.; and Itzfeldt, W.D. "User-Perceived Quality of Interactive Systems." IEEE Transactions of Software Engineering, Volume SE-4, July 1978.

FY 79-83 Research and Development Technology Plan, September 1977. R&D Technology Panel to the Mgt. Steering Committee for Embedded Computer Resources. Office of the Director of Defense Research and Engineering.

Gabriele, Gary; Jazayeri, Mehdi; and Mitchell, John. Requirements Formulation for MIS through System Sketching. Army Institute for Research in Management Information and Computer Science. Atlanta, Georgia.

Giese, Clarence Dr. "Partitioning Considerations for Complex Computer Based Weapon Systems." Journal of Systems and Software, January 1979.

H6180/Multics/URL User's Manual, Volume 1. ISDOS project, University of Michigan, Ann Arbor, MI. March 1977.

Hamilton, Margaret, and Zeldin, Saydean. The Foundations for Axes: A Specification Language Based on Completeness of Control. Charles Stark Draper Laboratory, Inc., Cambridge, Mass. Rev. October 31, 1975.

Hamilton, Margaret, and Zeldin, Saydean. "Higher Order Software - A Methodology for Defining Software." IEEE Transactions on Software Engineering, Volume SE-2, No. 1 March 1976.

Hamilton, M. and Zeldin, S. "The Manager as an Abstract Systems Engineer." Proceedings of COMPCON 77 Fall Conference, IEEE Computer Society, September 1977.

ICAM Program Prospectus, Air Force Materials Laboratory, Wright-Patterson AFB, Ohio; June 1, 1978

IORL Analyst User's Manual. Teledyne Brown Engineering, Huntsville, Alabama, 1977.

ISDOS Newsletter, Volume 11, No. 1, March 1979. The ISDOS project, University of Michigan, Ann Arbor, MI.

Infotech Programming Technology Product Description. Infotech International Incorporated, Pasadena, California, 1978.

Introduction to IORL. Teledyne Brown Engineering, Huntsville, Alabama, 1977.

Jones, Capers. "A Survey of Programming Design and Specification Techniques." Proceedings of Conference on Specifications of Reliable Software, IEEE Computer Society Catalog No. 79CHi401-9C, April 1979.

Macko, D.; Mesarovic, M.D.; and Takahara, Y. "Theory of Hierarchical, Multilevel, Systems." Mathematics in Science and Engineering, Volume 68. Cleveland, Ohio, Academic Press, 1970.

Murdick, Robert G., and Ross, Joel E. Information Systems for Modern Management. Second edition. Englewood Cliffs, New Jersey, 1975.

Nashman, A.E., and Gross P.D. A Digital Systems Development Methodology. Computer Sciences Corporation, Falls Church, VA 1978.

A Panel Session - "User Experience with New Software Methods." Proceedings of National Computer Conference, 1978. By Victor R. Basili, Chairman.

Pedersen Jan T., and Buckle, John K. "Kongsbert's Road to an Industrial Software Methodology." IEEE Transactions of Software Engineering, Volume SE-4, No. 4. July 1978.

Principato, Robert. A Survey of Specification Techniques for Parallel Processes. Charles Stark Draper Laboratory, Inc., Cambridge, Mass., January 25, 1977.

Problem Statement Language (PSL), Introduction and User's Manual. PSA Version 4.2 ISDOS Project, University of Michigan, Ann Arbor, MI, May 1977.

RDL/RDP User's Manual, Document No. RSS-060, Update D-4, Sperry-Univac, July 15, 1978.

Ramamoorthy, C.V. and So, H.H. "Software Requirements and Specifications: Status and Perspectives." Tutorial: Software Methodology, IEEE Computer Society Catalog No. EH0142-0, November 1978, Ramamoorthy, C.V. and So, H.H. "Survey of Principles and Techniques of Software Requirements and Specifications", Infotech State-of-the-Art Report in Software Engineering Techniques, 1977

Ramamoorthy, C.V. and Yeh, Raymond T. Tutorial: Software Methodology. IEEE Computer Society, Catalog No. EH0142-0, November 1978.

Ross, Douglas T. "Structured Analysis (SA): A Language for Communicating Ideas." IEEE Transactions of Software Engineering, Volume SE-3, No. 1, January 1977.

Ross, Douglas T. "Guest Editorial Reflections on Requirements." IEEE Transactions on Software Engineering, Volume SE-3, No, 1, January 1977.

Ross, Douglas T. and Schoman, Kenneth E.J. "Structured Analysis for Requirements Definition." IEEE Transaction on Software Engineering, Volume SE-3, No. 1, January 1977.

SCE/ISDOS User's Workshop, proceedings of. Held July 26-28, 1978. Irvine, California. ISDOS project, University of Michigan Ann Arbor, MI.

Santoni, P.A. "A Report on Three Methodologies/Tools for System Concept and Design." NELC Technical Note. Naval Electronics Laboratory Center, San Diego, CA November 17, 1976.

Slegel, Raymond C. Applying SREM to the Verification and Validation of an Existing Software Requirements Specification, Applied Physics Laboratory, John Hopkins University, Laurel, Maryland.

Stephens, Sharon A., and Tripp, Leonard L. "Requirements Expression and Verification Aid." Proceedings of Third International Software Conference, May 1978.

Taggart, W.M., Jr., Tharp, M.O. "A Survey of Information Requirements Analysis Techniques." ACM Computing Surveys, Volume 9, No. 4, December 1977.

Technical Report - SREM Evaluation Results Summary. McDonnell Douglas Astronautics Company - West, Huntington Beach, Califronia; October 1978. (Under U.S. Army Contract DASG 60-77-C-0009)

Teichroew, Daniel, and Hershey, Ernest A., III. "PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems." IEEE Transactions of Software Engineering, Volume SE-3, No. 1, January 1977.

Teichroew, Daniel. "Methodology for the Design of Information Processing Systems." Proceedings of Fourth Australian Computer Conference, Adelaide, South Australia, 1969.

Teichroew, Daniel; Hershey, Ernest Allen III; and Yamamoto, Yuzo. Computer-Aided Software Development, ISDOS project, University of Michigan, May 1977.

Uhrig, J.L. Software Engineering and Production Investigation. Bell Laboratories, Whippany, New Jersey, December 1, 1978.

Vick, C.R.; Scalf, J.E.; and McDonald, W.C. Distributed Data Processing for Real-Time Applications. Ballistic Missile Defense Advanced Technology Center, Huntsville, Alabama.

Willis, R.R. "DAS - An Automated System to Support Design Analysis." Proceedings of Third International Software Engineering Conference, May 1978.

Xerox Systems Methodology. Xerox Corporation, Rochester, New York, 1978.

Yourdon, Edward. "The Choice of New Software Development Methodologies for Software Development Projects." Proceedings of National Computer Conference, 1978.

Zave, Pamela and Fitzwater, D.R. Specification of Asynchronous Interactions Using Primitive Functions. Dept. of Computer Science, University of Maryland, College Park, Maryland.