ISSN 0105-8517

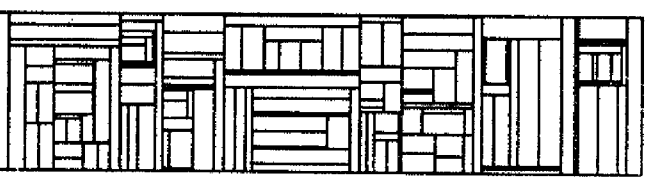
ISAC
— A Case Study of Systems Description Tools

Susanne Bødker
Jon Hammerskov

DAIMI PB-172
April 1984

To be presented at the Seventh Scandinavian Research Seminar on Systemeering, Helsinki, Finland, August 1984

Computer Science Department
AARHUS UNIVERSITY
Ny Munkegade – DK 8000 Aarhus C – DENMARK
Telephone: 06 – 12 83 55



Abstract.

ISAC[1,2] is one of many methods for systems development. Such methods are developed in order to improve the quality and efficiency of systems development processes. Systems descriptions are made and used throughout the systems development process. The tools used for systems descriptions have an important impact on the quality of the descriptions which are produced and hence on the quality of the product of the systems development process, the computer based system.

This paper presents an analysis of the systems description tools used in ISAC by presenting some problems that occured in an actual systems description process.

The main problems are that ISAC imposes a structure on the systems descriptions which makes it impossible to describe important properties of information systems properly, moreover, other important properties cannot be described at all. ISAC can only be used for sketching properties that are already familiar to the user, thus an ISAC description is insufficient as a means of communication within the systems development process.

Why ISAC?

When we first got interested in ISAC we discovered that ISAC is widely used throughout Scandinavia, especially for teaching purposes. To find out why and how, we interviewed various practitioners and teachers of ISAC and we started to use ISAC ourselves.

The main purpose of our work, however, is identical to the purpose of developing methods for systems development: improving the quality of systems development. A necessary step to fulfill the aims of developing better methods is to gain a better understanding of practical problems and conflicts in systems development[3]. As systems descriptions are made and used for analysis, design decision and communication purposes throughout the systems development process, the systems description process is perhaps the most important subtask within the systems development process[4].

In the work, on which this paper is based, we encountered several problems with the ISAC tools and techniques. By presenting these problems here we hope to provide parts of a basis for the definition of better systems description tools than the ones defined in ISAC. This is definitely more important than just pointing out the deficiencies of ISAC, even more so as we found similar problems in other methods such as HIPO[5] and SA[6].

In order to improve the reader's understanding of the tools we first give a brief presentation of ISAC. This presentation is followed by a presentation of our case study: the description process in which we took part, and the analysis of ISAC.

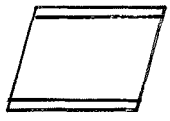
ISAC.

Lundeberg et al.[1,2] presents ISAC as a method intended for information systems development. An information system is a system that has been developed to create, collect, store, process, distribute and interpret information. ISAC is intended to be used when several people, including the users, are going to take part in the systems development process. Users means management since ISAC does not deal with influence to interest groups such as local trade unions.

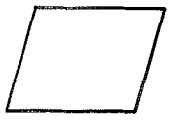
The guidelines, prescribed by ISAC, for the systems development process can be characterized as a set of tools and techniques since ISAC does not include guidelines for organizing the systems development process.

The main technique is devided into five phases: change analysis, activity studies and information analysis which are called the problem oriented parts and data system design and equipment adaption, called the data oriented parts. The change analysis focuses on the organization as it exists whereas the other four phases are concerned with design and realization of the new information system.

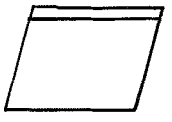
In the following, we shall study the problem oriented parts of the method in which the users and their needs and problems are in focus[1], the purpose of this being to study the ability of the ISAC tools to act as construction and communication tools.



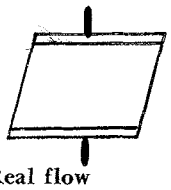
Real set
Set of persons and/or material.



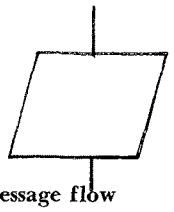
Message set
Set of messages, e.g. documents or information by telephone.



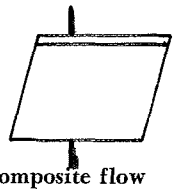
Composite set
Set comprising persons/material as well as messages.



Real flow
Flow of persons/matrial only.



Message flow
Flow of messages.



Composite flow
Flow of persons/material as well as messages.

Activity
People and other resources take part in the activity.

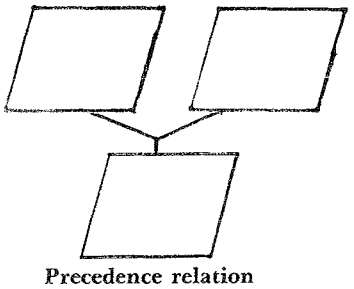
All flows are assumed to go from top to bottom on the graphs.
Arrows are needed on upward and (possibly) horizontal flows only.

figure 1. Legend for A-graphs.

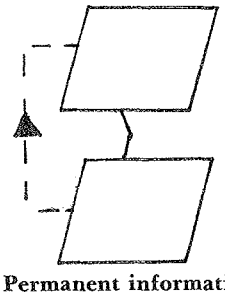
During the change analysis, the activities of the organization as it is are described in order to identify the changes that have to be made through the introduction of the information system. During the activity studies these changes are described as new or changed activities within the organization. The main technique in both phases is the A-graph (activity graph) technique. During the information analysis the information system is described. From the results that are going to be produced by the information system, viewed as information sets, all precedence relations are described using I-graphs (information precedence graphs). The structure of the information sets is described using C-graphs (component graphs). Finally the information processes, identified at the bottom of the I-graph hierarchy, are described.

We will base our discussions in the following on the A- and I-graph tools only, because A- and I-graphs are the general tools in the problem oriented parts. They are the tools that tie together the work processes prescribed, and they are the only tools for which we have been able to find a somewhat formalized description * within the ISAC literature.[7,8] The legends of the symbols used in A- and I-graphs are shown in figure 1 and 2, and figure 3 shows an example of an A-graph (see also Lundeberg et al.[1,2]).

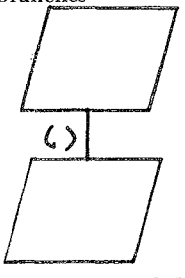
Information set
An information set consists of a number of messages. An information set is described by the information content in the message.



Precedence relation
Relations between information sets. The pecedence relation is illustrated by a node and a number of branches



Permanent information set
An information set which exists in different generations, i.e. with a simmilar structure but with information contents from different points of time.



Fictitious precfedence relation
Two identical information sets are shown by a fictitious pcedence relation, indicated by the parentheses without a reference code.

Predecessors are placed above successors.
Arrows are needed only if relation is reversed.

figure 2. Legend for I-graphs.

* A formal definition of the syntax and semantics of A- and I-graphs can be found in our thesis.[9]

The systems description process.

We initially became aware of the problems in ISAC when we used the method in a project about tools for systems description. In order to use ISAC as an example for our master thesis work we settled for a description of a research library, because we were familiar with the activities going on in this library and it seemed that an introduction of an information system in the library would fit the area of application of the ISAC approach.

We worked our way through the systems description process using blackboard, chalk, pen and paper and a collection of ISAC literature. Throughout this process, we recorded the problems encountered, no matter how big or small they seemed.

Two months of work by two persons resulted in a description with 50 A- and I-graphs arranged in a hierarchy with 8 levels of graphs. We did 2-3 iterations and due to the hierarchy a lot of time was spent keeping the graphs up to date: a change in one graph would often cause changes in five other graphs. Out of four man months we spent approximately 75% of the time on redrawing, rewriting, discussing how to draw the graphs technically correct, and on checking consistency among the graphs. Only 25% of the time was spent on reaching a mutual understanding of the library and of the design of the library information system.

We found a number of technical problems. One is that the number of activities in one A-graph is limited to 9 and the number of ingoing or outgoing sets for an activity is limited to 6 (similar for I-graphs). Every now and then we would reach these limits and the only solution would be to introduce an extra level of graphs or to unite two sets into one, even though this conflicted with the structure of the properties that we were about to describe. However, many of the problems encountered were more serious than this. We will give examples of these problems in the following.

Experiences with ISAC.

ISAC claims to describe information systems, but we have, through this study, found that several properties of information systems cannot be described properly. These properties can be categorized in the following way:

properties the structure of which conflicts with the structure of the description,

properties that cannot be described at all due to the lack of certain concepts in ISAC,

properties the structure of which conflicts with the systems description process.

We are going to compare the language concepts of ISAC with concepts of ordinary programming languages because the programming language world provides us with an appropriate conceptual framework and because ISAC descriptions are going to form the prerequisite of programming.

The following presents examples of the first category:

When describing a library an important concept is the borrower. However, the classification of a borrower is very difficult in ISAC: On one hand the borrower is part of the flow in the library, i.e. the library as an activity takes the borrower as part of its input flow and does something to her (e.g. supplies her with some books). On the other hand the borrower is an active component who performs various actions (e.g. removes a book from the stock of books). In ISAC, the borrower's actions must be identified as activities, whereas the borrower as an information carrier must be characterized as a set. *There is no way this concept can be described as a whole.* In our description, one part of the description of the borrower is refined in one A-graph, perhaps

together with the refinement of a part of the description of some other concept, whereas other parts of the description are refined in another A-graph. *The description of the borrower cannot be refined or decomposed into meaningful components.*[10] The definition of the borrower is scattered all over the description of the library. A consequence of this is that *there is no way to define the concept borrower once and for all as a primitive and then refer to this definition later on.*

If we compare this to the area of programming languages, ISAC would be a language without types and subroutines and there would be no way to have classes, as SIMULA[11] calls them or similar abstraction facilities. For a discussion of the need for such concepts we refer to Horning.[10]

The following examples belong to the second category:

One of the other important concepts of the library is the stock of books. We have classified this as a set but we know that the concept stock covers many characteristics: the stock is both the total collection of books in the library and the way the books are organized in the library: they are organized in sections according to type, title, author, etc. The books have relations among them - e.g. one can talk about the place of a book even though the book itself has disappeared. ISAC doesn't provide us with the tools to describe this. *There is no way we can talk about the stock as an ordered set of books and about the relations among the books.* Also, we are not able to describe the algorithms for finding the "right" book in the stock, since *the description of an activity does not include algorithms.*

In the world of programming languages this compares to a language without structured data objects and without the possiblility of specifying statements, expressions or similar concepts.

In our example, when the librarian receives a new book to be included in the stock of books this procedure is followed: All books (but not periodicals) are manually numbered. This number is unique except when the book is a replacement for a lost book. Information about the book is registered in one or more files depending on whether it is a replacement, a periodical, a bibliography, etc. The consistency and integrity of the files are maintained by restricting the order in which each step of the registration procedure is executed. Such restrictions cannot be captured in ISAC. *The only thing one can do is to increase detail, but detail is not equivalent to precision,*[12] since detail has to do with the structure of the perception process and precision with the structure of the properties comprehended.

It is important to note that such restrictions are described in the information process analysis. The implication then is that knowledge of this kind must be remembered and communicated outside the description from the initial descriptions (A- and I-graphs) to the information process analysis.* The A- and I-graphs are however intended to be complete descriptions - that is, they are not supplemented by other descriptions. Therefore they should contain all necessary knowledge of the system described.

However, all presentations of the ISAC method contains A- and I-graphs that are commented in prose. Careful study of these examples reveals that important aspects of the system described in the prose cannot be inferred from the graphical descriptions alone.

As an example of this we reproduce an A-graph from Lundeberg et al.[1] - shown in figure 3. In order to concentrate on the information contents of this description we also give an equivalent textual description, much like a programming language - shown in figure 4.
As with any other presentation of ISAC Lundeberg et al.[1] give an interpretation of the graph in ordinary prose, saying:

---

* Reported experience with the use of ISAC supports this observation.[13]

Also, for the design of computer based systems the lack of information in the descriptions is crucial. In our example several questions of importance cannot be answered by examining our description, e.g. can a book be registered as borrowed without being registered in the authors catalogue?

activity TRACO
  inputs
    Material Delivery : *composite*
    Error Report : *information*
    Personnel : *composite*
    Equipment to be Maintained : *material*
  outputs
    Material Order : *information*
    Maintained Equipment : *material*
  is
    activity Maintenance Planning
    inputs
      Error Report : *information*
      Maintenance Experience : *information*
    outputs
      Material Request : *information*
      Work Order : *information*
  and
    activitity Material Provission
    inputs
      Material Delivery : *composite*
      Material Request : *information*
    outputs
      Material Order : *information*
      Material : *material*
  and
    activity Direct Maintenance
    inputs
      Personnel : *composite*
      Equipment to be Maintained : *material*
      Material : *material*
      Work Order : *information*
    outputs
      Maintained Equipment : *material*
      Maintenance Experience : *information*

figure 4. Textual activity description.

The third category contains the following examples:

The A- and I-graph tools are intended to support a top-down stepwise refinement process which gives the description *one* hierarchical structure. This means that two acitvities, once identified and separated, can never be related to each other again and the relation described. We would have liked to refine some critical parts of our description, which have relations among them - e.g. "borrow a book" and "send request to borrower to return book". In the description, however, the two parts are placed in different parts of the hierarchy, a fact that makes it impossible to refine these activities without refining other as well and to describe relations between the two.

Furthermore, it is impossible for us to describe many dependencies, especially among various files and catalogues. The following dependencies are very important in the library: there is at most one borrower per book, a book that is missing cannot be borrowed, neither can it be found on a shelf. *There is no way we can describe these non-hierarchical dependencies in ISAC.*

This supports the claims of Jackson[15] and others who say that top-down is not a technique to be used when developing programs and systems, when solving problems or when perceiving or designing anything.

type QUEUE **is:**

NEW:		->	QUEUE
ADD:	QUEUE x ITEM	->	QUEUE
FRONT:	QUEUE	->	ITEM
REMOVE:	QUEUE	->	ITEM
IS.EMPTY?:	QUEUE	->	ITEM

figure 5. Abstract data type QUEUE.

Summary.

Systems descriptions are used throughout the systems development process, both in order to construct the computer based system and in the communication about computer based systems among the different parties involved in the systems development process.

A person who is using ISAC as a systems description tool will not necessarily be able to reflect her perception of the new computer based system in the description. The reason for this is the lack of precision in the description and the structure which is imposed on the concepts by the tools. Thus, the description will be inadequate as a foundation for the detailed construction of the computer system (i.e. programming) and as a means of communication.

To improve ISAC we would need a way of describing activities which is more than just detailing, but this leaves basic problems that are not so easy to solve: the problem of being able to capture the concepts needed in the description and the problem of handling the complexity of the description. This leads to the conclusion that ISAC is useful when communicating an overview of the flow of data in a system, but descriptions using other tools must be added in order to make a precise description and in order to make descriptions of other aspects than data flow.

References:

1. **M. Lundeberg, G. Goldkuhl & A. Nilsson**: *A Systematic Approach to Information Systems Development — I. Introduction.*
Information Systems vol.4 pp.1-12, 1979.
2. **M. Lundeberg, G. Goldkuhl & A. Nilsson**: *A Systematic Approach to Information Systems Development — II. Problem and Data Oriented Methodology.*
Information Systems Vol.4 pp.93-118, 1979.
3. **L. Mathiassen**: *Systemudvikling og systemudviklingsmetode* (in Danish - *Systems development and systems development method*)
DAIMI PB-136, Århus 1981.
4. **K. Nygaard**: *Delta prosjektet og dets tilknytning til problemene i systemutvikling* (in Norwegian - *The Delta project and its relation to the problems in systems development*)
in DAIMI PB-46, Århus 1975.
5. **H. Katzan Jr.**: *Advanced Programming, Programming and Operating Systems.*
Computer Science Series, Van Nostrand Reinhold Co. 1970.
6. **T. DeMarco**: *Structured Analysis and System Specification.*
Yourdon Inc., Prentice-Hall 1979.
7. **M. Lundeberg & E.S. Andersen**: *Systemering — Informationsanalys* (in Swedish -*Systemeering - Information Analysis*).
Studentlitteratur, Lund 1974.
8. **H. Nissen & E.S. Andersen**: *Systemering — Verksamhetsbeskrivning* (in Swedish -*Systemeering Activities Description*).
Studentlitteratur, Lund 1978.
9. **S. Bødker & J. Hammerskov**: *Grafisk Systembeskrivelse* (in Danish -*Graphical Systems Description*).
Thesis, University of Århus, DAIMI IR-33,34,35 Århus 1982.
10. **J.J. Horning**: *Some Desirable Properties of Data Abstraction Facilities.*
Sigplan Notices, Vol.8 No.2 1976.
11. **O.-J. Dahl, K. Nygaard & B. Myhrhaug**: The SIMULA 67 Common Base Languages.
Norwegian Computing Centre, Oslo 1968.
12. **A. Munk-Madsen**: *Systembeskrivelse med brugere*(in Danish - *Systems Description with Users*).
Thesis, University of Århus, DUE-notat 9, Århus 1978.
13. *Systematisk verksamhetsbeskrivning och informationsanalys enligt ISAC — några erfarenheter* (in Swedish *Systematic activity-description and information analysis according to ISAC — some experiences*).
Studentlitteratur, Stockholm 1979.
14. **J. Guttag**: *Abstract Data Types and the Development of Data Structures.*
CACM Vol.20 No.6, June 1977.
15. **M. Jackson**:
Talk by Michael Jackson in Copenhagen, May 1981.