# Information Systems Development
# A Frame Of Reference And Classifications

**Anders G. Nilsson**

**The Institute for Development of Activities in Organizations**
**Institute V**
**Box 6501**
**S-113 83 STOCKHOLM, Sweden**

**Abstract:** This paper is an attempt to classify the process of information systems development in different ways. The main perspective is that information systems should support business activities in organizations. The development process is illustrated by our V-model, ISAC-model and SIV-model. We have experience of applications of these models from Swedish industry. At the end a three-dimensional classification in function analysis, object analysis and event analysis is presented as a base for comparisons of information systems development methodologies.

**Keywords:** Application package, classification, computer support, development model, function/object/event analysis, information systems development, life cycle, methodology, transformation.

**Please note:** This paper is related to the report by Bernt T. Boström "Information Systems Development: Supporting Methodologies With Computerized Tools".

# 1 Information Systems Should Support Business Activities

A company is running some kind of business activities. The direction and scope of the activities are guided towards some business goals of the company. We can classify the activities in different ways:

- operative or administrative activities
- manual or automatable activities

The *operative* activities realize the company's business goals. They produce the products and services for customers on the market. Operative activities are the real business activities in a company. *Administrative* activities are supporting activities. They control and coordinate the operative activities in order to fulfil the business goals. The administrative activities support operative activities with appropriate information. Information systems are examples of administrative activities.

Both operative and administrative activities can be executed in different ways. When people execute the work tasks, we have *manual* activities. If machines can execute the work, we have *automatable* activities. An example of this is computer-based activities. We can combine operative/administrative and manual/automatable parts in an activity matrix for a company. See figure 1 with an example from a car repair company.

| EXECUTION<br>FUNCTION | MANUAL | AUTOMATABLE |
|---|---|---|
| OPERATIVE | CAR INSPECTION | ENGINE DIAGNOSTICS |
| ADMINISTRATIVE | WORK SHOP PLANNING | INVOICING |

Figure 1 Activity matrix: car repair company

The main perspective we have here is that information systems should support the business activities in a company. The information systems should provide the desired effects and benefits for operative activities. This is a pragmatic and praxiologic view of information systems [8,12,13]. We are against a perspective saying that information systems are ends in themselves. This view leads to a separation between information systems and business activities which is purposeless.

Our perspective admits that a specific information system can support other information systems before giving direct effects to the operative activities. A trend is to distinguish many small information subsystems each giving support to a well defined local activity (function), rather than to find a totally integrated information system in a company.

An *information system* consists of three main parts; input messages, message processing and output messages. We have also processing rules which control the execution of the information system. If the processing rules are formalized we can have computer-based information systems. But if the processing rules are "fuzzy" and we need a lot of personal knowledge, judgement and intuition, the information systems must be manual. A purposeful information system shall help users to make good decisions and support their actions.

## 2 Life Cycle Of Information Systems

Information systems, like other products, are going through a *life cycle*. We can identify four main phases in a life cycle for information systems:

1 Information Systems Development

2 Information Systems in Use (Operation)

3 Information Systems Maintenance Management

4 Information Systems Withdrawal

There is an interaction between these phases. See figure 2. We start with information systems *development*. This means work with analysis and design of information systems. We create or acquire information systems which shall fulfil user needs and requirements. After that it is time for the users to begin to utilize the information systems in their activities. The information systems are now in *use* or *operation*.

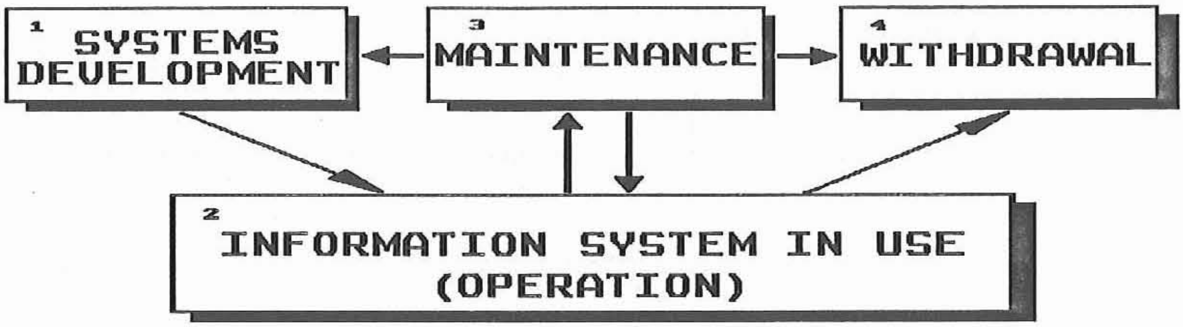


Figure 2 Life cycle

After some time we need to *maintain* the information systems. Maintenance consists of measures for corrections, adaptations, improvements and reorganizations of information systems. The causes for maintenance can be changes in business activities, in the environment of the company or in information technology. Maintenance management is taking place when information systems are in use (operation).

An important task for maintenance management is to make regular follow-ups (reviews, audits) of the quality of information systems; at least once a year. A follow-up can lead to different outcomes:

- status quo
- maintenance measures
- renewed systems development
- need for withdrawal

Information systems will at the end be obsolete or unprofitable. They will not give the desired support for the business activities in the company. The last phase of the life cycle is therefore *withdrawal* of information systems. This can be the starting point for a new life cycle.

# 3 Classification Of Information Systems Development

We shall now concentrate on the process of information systems development. But development of information systems is only one possible strategy or measure to improve business activities in a company. Examples of other forms of corporate development are product/service development, market development and organization development. In a real case you often choose a combination of development measures in order to satisfy the business goals of the company. This points out the need for a *business activity analysis* as a starting point for all development work. The business activity analysis can lead to different outcomes:

- information systems development
- other development
- status quo (current situation is sufficiently good)
- liquidation (of activities that are obsolete)

We focus our interest on information systems development. The business activity analysis must have pointed out that information systems development is a possible and appropriate strategy for improvements in the company.

Information systems development can be accomplished in different ways. We need to characterize the process of information systems development into some appropriate situations. A frame of reference will be presented in this sense. See figure 3. We first classify information systems development after the degree of "*prespecification*" into:

1 *acquisition of standard application packages*
development with help of ready-made subsystems

2 *use of application generators*
development with help of semi-manufactures built upon some kind of 4GL-tool

3 *specification and design of tailor-made systems*
development with help of new specifications from scratch

Each of these strategies represent a very special development milieu. They require quite different methodologies in order to accomplish the development work in a fruitful manner. Information systems development can also be regarded as an abstract or a concrete process by users. This leads to another way of classifying information systems development after the degree of "*concreteness*" into:

1 *analytical design*
an abstract model of the information systems is analyzed in detail, the requirement specification is "frozen" before an implementation takes place

2 *experimental design (prototyping)*
a simplified prototype (experiment system) of the information systems is quickly implemented. The users get a concrete picture and can define their information needs in a better way before a final implementation

Information systems development is always a change process. There is a need for a cooperation between users and systems analysts. We can classify information systems development after the degree of "*user influence*" on the process into:

1 *expert work*
systems analysts are regarded as experts and develop information systems on their own for the users (expert strategy)

2 *collaborative work*
users and systems analysts develop information systems in collaboration within teamworks (anchoring strategy)

3 *user work*
users develop information systems on their own (self-service), if needed the users can get help from the systems analysts during the development process (process strategy)

Above we have classified information systems development after the degree of prespecification, concreteness and user influence. They are independent classifications. We can therefore combine the different outcomes into 3 x 2 x 3 = 18 different situations of information systems development. In the past the dominating combination was "tailor-made systems with analytical design by expert work". In

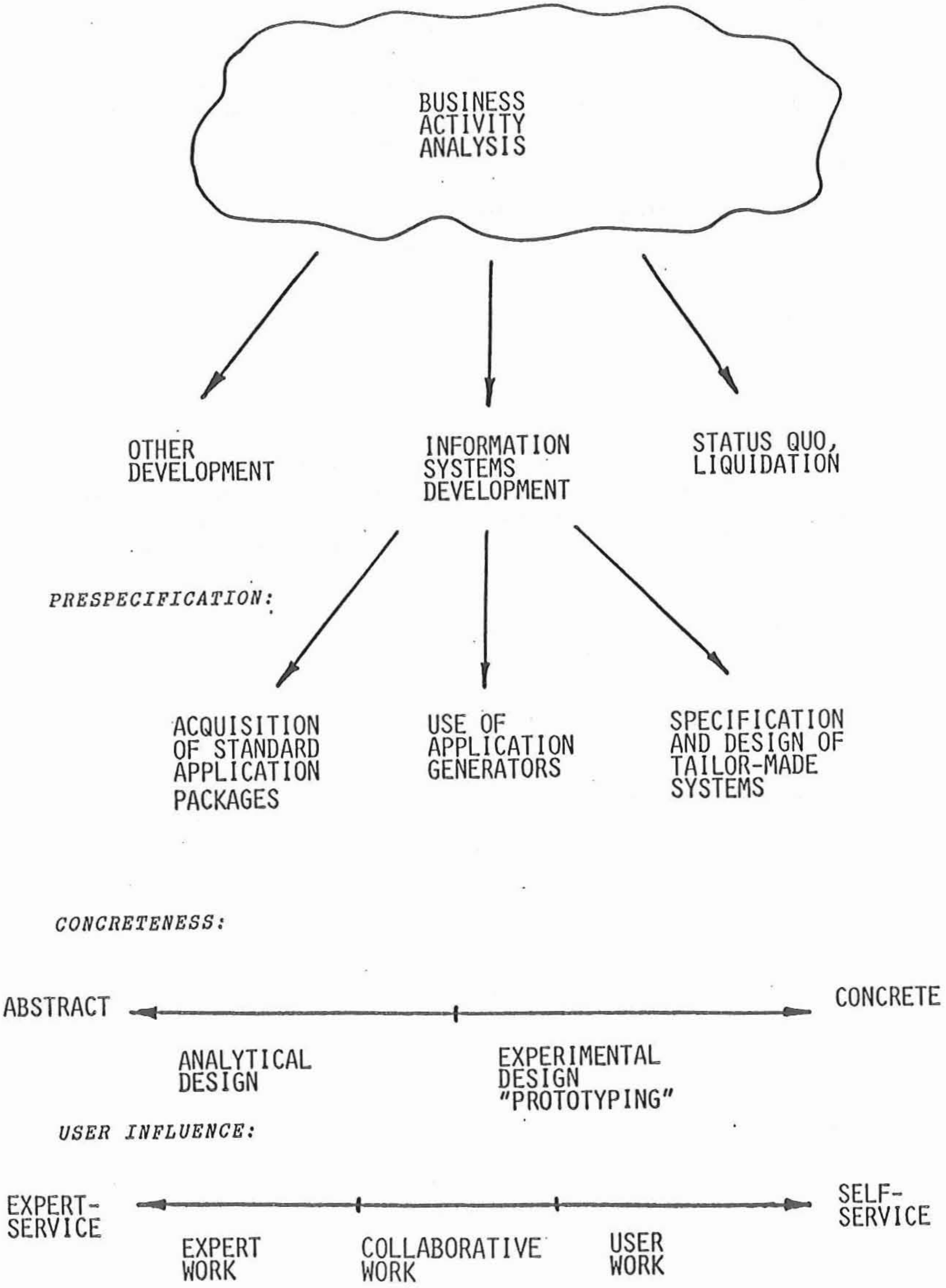

BUSINESS ACTIVITY ANALYSIS
OTHER DEVELOPMENT
INFORMATION SYSTEMS DEVELOPMENT
STATUS QUO, LIQUIDATION
PRESPECIFICATION:
ACQUISITION OF STANDARD APPLICATION PACKAGES
USE OF APPLICATION GENERATORS
SPECIFICATION AND DESIGN OF TAILOR-MADE SYSTEMS
CONCRETENESS:
ABSTRACT
CONCRETE
ANALYTICAL DESIGN
EXPERIMENTAL DESIGN "PROTOTYPING"
USER INFLUENCE:
EXPERT-SERVICE
SELF-SERVICE
EXPERT WORK
COLLABORATIVE WORK
USER WORK

Figure 3 Classification of information systems development

the future we can have many more situations than we have sketched here. But a common denominator is the need for a business activity analysis before we choose our strategy for information systems development.

# 4 What Is a Methodology?

The concept of methodology for information systems development has been debated during a long period of time. Today most researchers agree upon the fact that a methodology gives a set of guidelines for a systematic way of working with development tasks. It is important to realize that every *methodology* is built upon:

1 a *perspective* including principles and values for good information systems and systems work

2 a set of more or less defined *concepts* which the development work is based upon

Information systems development is a comprehensive and complex work task. You have to tackle many problems and to decide upon many questions. Therefore a methodology have to divide the development work into a number of perceivable phases. A useful *methodology* should consist of the following parts:

3 a *model* with a number of development areas (phases)

4 a *method* with a number of coherent work steps (method steps) for each development area

5 a set of *description techniques* for actual document types produced during the work steps

6 a set of *tools* like computer supports for document handling and to manage transitions between work steps

In a real case you often have to choose between alternative methods, techniques and tools. Different situations may require different methodologies. We have also a possibility to combine some methodologies with each other and get a "methodology chain". Sometimes we use the concept *approach* as a synonym for methodology. But an approach could be a whole methodology or only some parts of it.

# 5 Development Models - Our Perspective

The development process will now be illustrated by some models from which we have derived experience from applications in Swedish industry. They are:

1 the *V-model*
an overall model for corporate development

2 the *ISAC-model*
a general model for information systems development

3 the *SIV-model*
a specific model for acquisition of standard application packages

We also present the history and evolution of our development models with an emphasis on the ISAC-model.

## 5.1 V-model

The V-model is focused on corporate development with a connection to information systems development. It gives a broad perspective for development of business activities in organizations. The model has an ability to comprise different methods, techniques and tools. Our V-model consists of nine development areas (see figure 4):

V1 *Business Diagnosis*
a short and rapid diagnosis of the company's business activities; this gives direction and scope for further development work

V2 *Change Study* (Change Analysis)
a deeper and more detailed study of problems, needs for changes and alternative solutions for a delimited activity area in the company; this leads to a change programme

V3 *Activity Study*
a study of information systems and their potential contributions to the business activities, we classify the information systems in manual and computer-based parts

V4 *Information Study* (Information Analysis)
a detailed study of the user's information needs; this leads to a requirement specification for each information system with messages and processes

V5 *System Design*
a technical design of data systems following the requirement specification; we create data/program structures and manual routines

V6 *Realization*
we build and manufacture the information systems according to the system design, such as program coding, data base organization and system tests

V7 Implementation
means that the users start to utilize the developed or acquired information systems in their activities

V8 *Follow-up* (Post Audit)
we summarize experiences from using the information systems; achieved results are checked against the change programme

V9 *Business Assessment*
we check that the developed activity areas correspond to our visions from business diagnosis

There is a logic that we draw the model as a "V". The development areas has a specific correspondence to each other. Business assessment is a check to business diagnosis, follow-up is a check to change study, implementation shall fulfil the results from activity study and finally realization shall fulfil the results from information study.

# V - MODEL

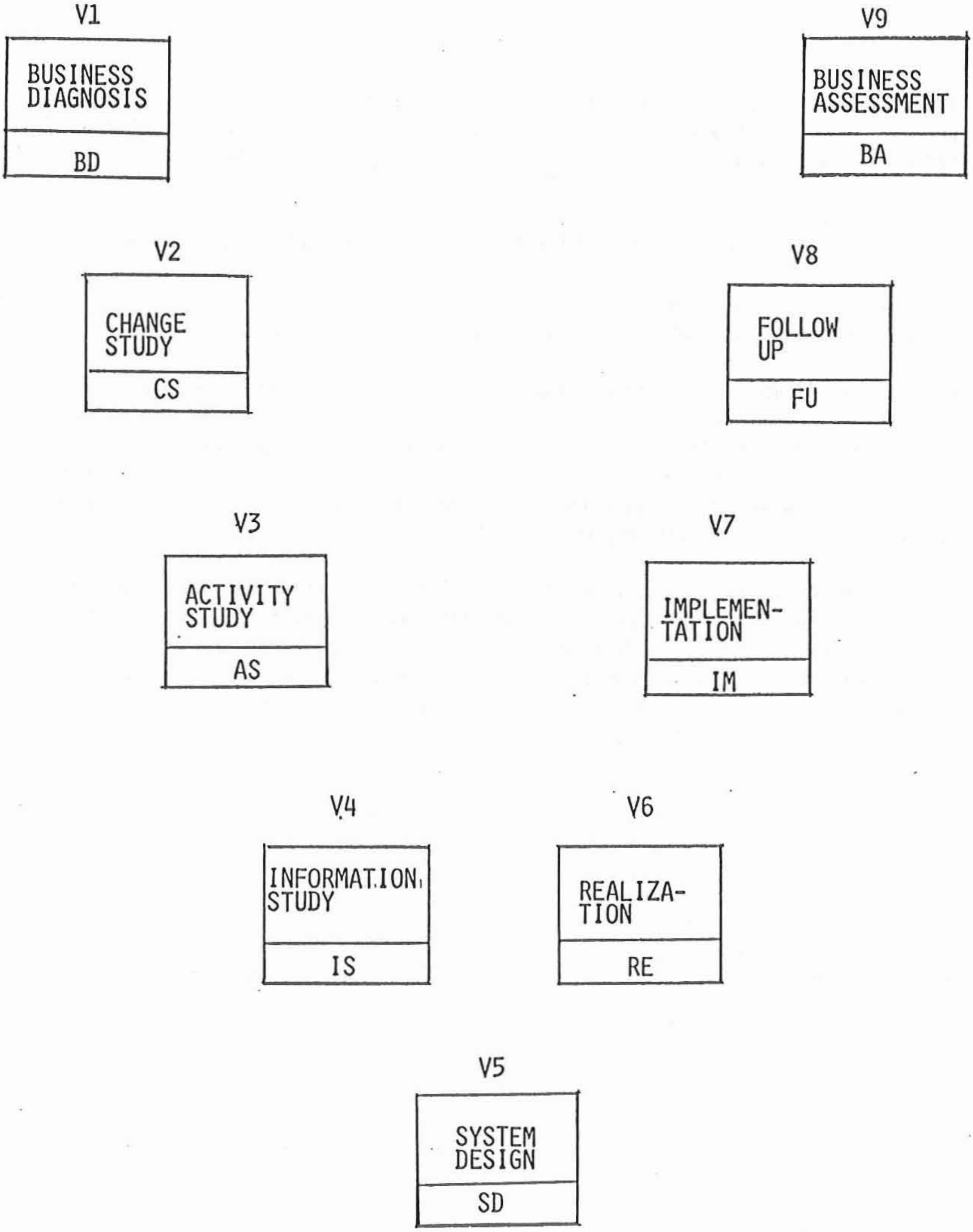


Figure 4 V-model

We have used the concepts business activity analysis and information systems development many times. Business activity analysis corresponds to the development areas V1, V2 and V3 in the V-model. Information systems development corresponds to the areas V3, V4, V5 and V6. Two important features with the V-model are the possibilities to use the ISAC-approach (with computer support GraphDoc) and the SIV-approach (with computer support SIV-DOC).

## 5.2 ISAC-model

The ISAC-model is general for information systems development with an emphasis on tailor-made and generated systems. ISAC is an acronym for "Information Systems work and Analysis of Changes". The ISAC-model consists of two main development areas according to the acronym:

1 *Change Analysis*
investigation of problems and possible solutions for business activities in a company

2 *Information Systems Development* ("Systemeering")
analysis and design of information systems in order to support the business activities

The ISAC-model corresponds to the development areas V2, V3, V4 and V5 in the V-model. The ISAC-approach is a complete methodology and it is published in a full version [14] and a short version [3]. For a summary of the short version, see figure 5. The most interesting feature of the ISAC-methodology is the use of activity descriptions with A-graphs (activity graphs). The A-graphs show how information systems support business activities in a company. A-graphs is a part of the SDA-technique for Systematic Description of Activities.

GraphDoc is a powerful tool for drawing and updating A-graphs (and I-graphs). The computer support guarantees a consistent documentation. GraphDoc can be used during the development areas V1, V2, V3 and V4 in the V-model. In the ISAC-model GraphDoc is used during work steps with graph drawing. There have been many attempts to produce computerized tools for the ISAC-methodology, but GraphDoc is the first really useful product in this sense. An interesting trend is the use of knowledge based systems (expert systems) as computerized tools for the ISAC-methodology [11].

## 5.3 SIV-model

The SIV-model is a specific approach for acquisition of standard application packages to support business activities. The acronym SIV stands for "Standard application packages for Improved business Activities ( = "Verksamhet" in Swedish)". Application packages are more and more taking over the importance of self made systems [6,9]. The SIV-model consists of three main development areas (see figure 6):

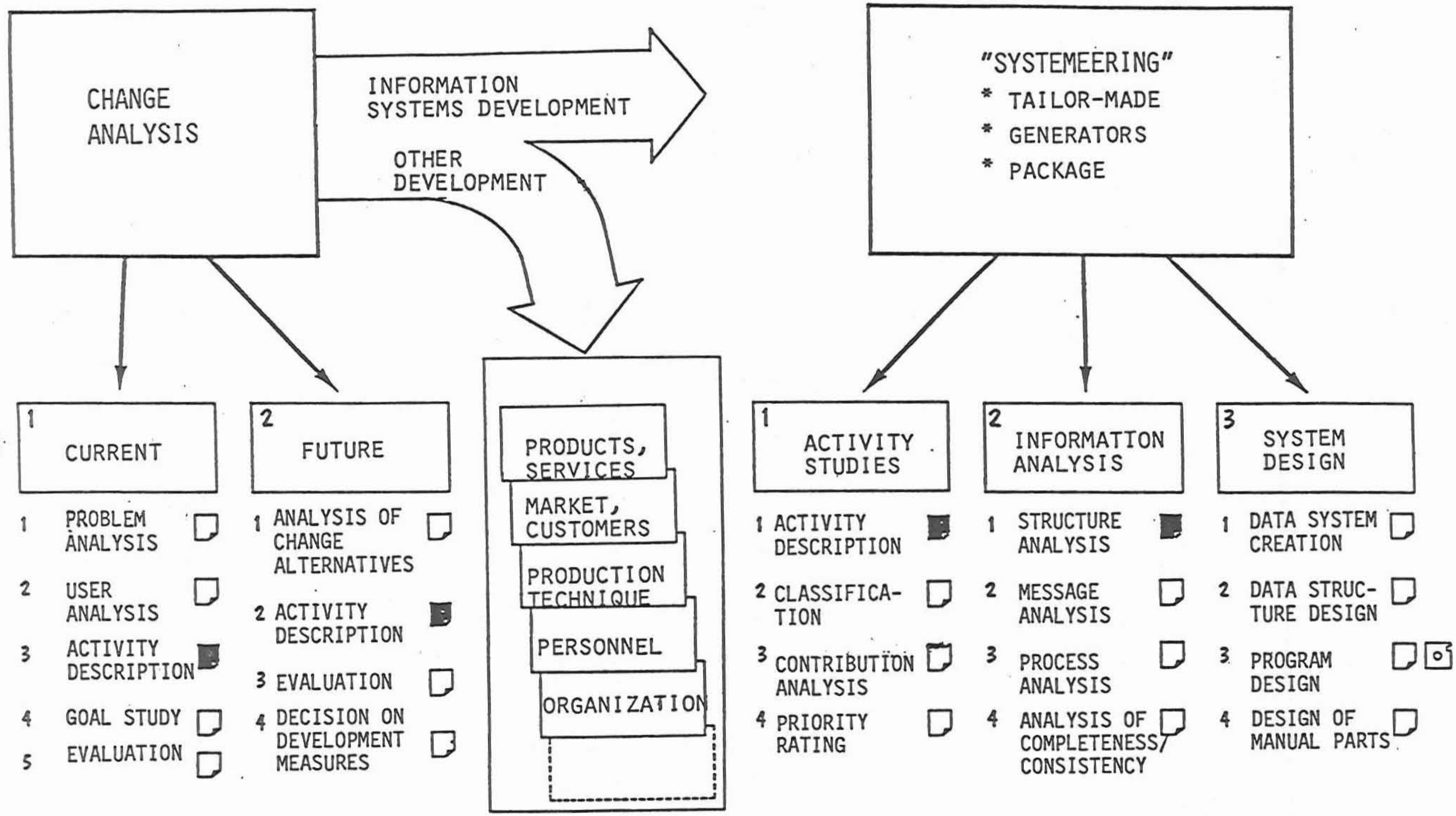


Figure 5 ISAC-model

1 *Choice*
between several available packages on the market for our delimited activity area (V3 in the V-model)

2 *Adaptation*
of the application package and the business activities to each other (V4, V5 and V6 in the V-model)

3 *Installation*
of the application package in real life to support the business activities (V7 in the V-model)

The SIV-model comprises the development areas V3 through V7 in the V-model. A change study precedes and a follow-up succeeds the SIV-model. The SIV-approach is published as a complete methodology [1,2]. It has a customer perspective when dealing with application packages towards a vendor.

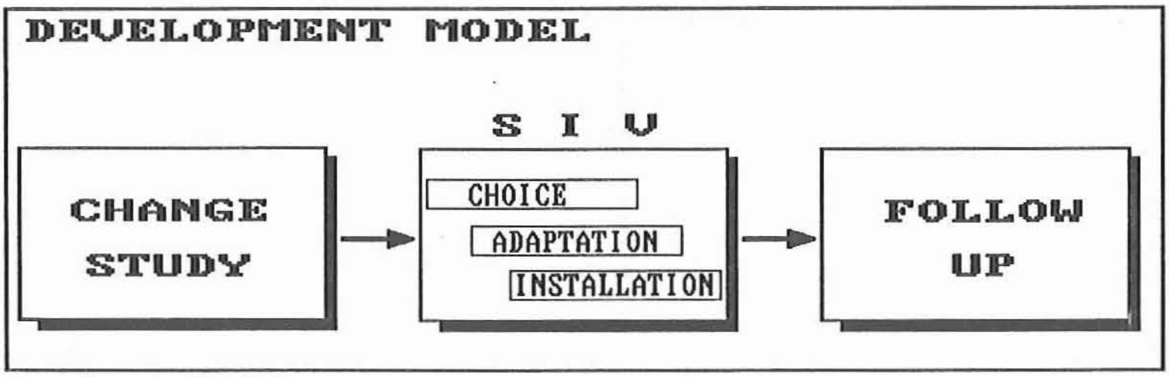


Figure 6 SIV-model

A special thing in this development milieu is that we have to make comparisons between our requirements for the business activities and the properties of the application packages. In this sense we need access to three types of documentation; activity description, package description and a compared description. See figure 7. Activity descriptions with A-graphs can be a powerful basis for both choice and adaptation of application packages.

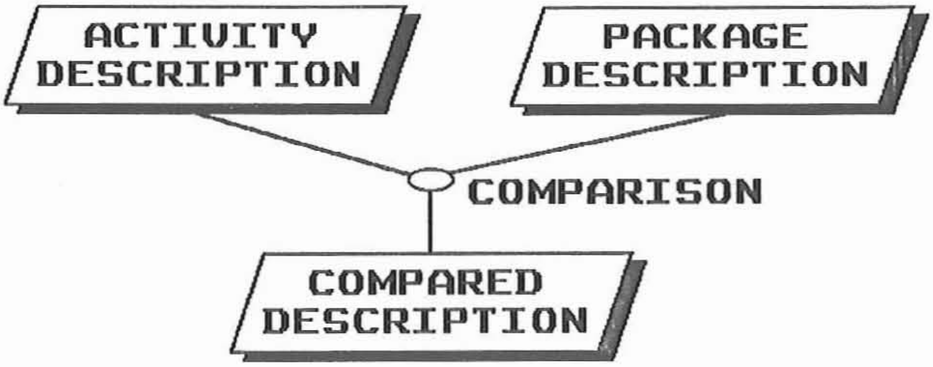


Figure 7 Documentation (according to SIV)

SIV-DOC is a computer support for applying the SIV-approach in a more efficient manner. The tool is for the moment only a prototype in order to illustrate a choice of application packages on the market. A user defines a search profile for his demands and then scans through databases of application packages. SIV-DOC can be integrated with the tool GraphDoc and corresponds to the development areas V3 and V4 in the V-model.

## 5.4 Evolution

We will now present the history and evolution behind the ISAC-approach. See figure 8. Information systems development starts with some perceived problems of the users and goes through a requirement specification to implementation of computer solutions. The first version of the ISAC-approach was developed in 1968. The *information analysis* area was invented as a purposeful basis for *system construction*. At this time many systems were built without regarding the real information needs of the users. Information analysis became an instrument for doing better computer programs.

Empirical tests of the ISAC-approach showed that information analysis was not enough. We have to know if the information systems are a real support for the business activities. So around 1972 the development area *activity analysis* was invented. Further empirical tests of the ISAC-approach showed that even activity analysis was not enough. We have to know if information systems are solutions to the existing problems or if other development measures should be taken. So around 1976 the development area *change analysis* was invented. At the same time system construction was divided into *data system design* and *equipment adaptation*. The ISAC-approach now became a complete methodology and was published in a book [14].

After starting up Institute V during 1981 we have done some more empirical tests. Around 1982 we divided change analysis into two subareas; *business diagnosis* and *change study*. The business diagnosis is a broad change analysis for the whole business, while the change study is a delimited analysis for some activity area. After a change study you can choose between *tailor-made*, *generated* or *package systems*. All of this we try to take care of in the V-model. The SIV-model is an extension for dealing with application packages.

# 6 Classification Of Methodologies

Many researchers have done a lot of work to classify different information systems development methodologies [5,7,17,18,19,20]. A common approach is to define a broad range of evaluation criteria for comparison of methodologies. It could be over hundreds of criteria to handle in a concrete evaluation situation. There is a need for a more comprehensive view of the available methodologies on the market today. We have for some time looked for a simple, complete and fruitful classification of methodologies. A proposal is to focus information systems development on three different aspects [15,16, 22]:

| | | | |
|---|---|---|---|
| 1 | Function analysis | (process-driven perspective) | (late 1960's) |
| 2 | Object analysis | (data-driven perspective) | (mid 1970's) |
| 3 | Event analysis | (behaviour-driven perspective) | (early 1980's) |

ISAC - APPROACH

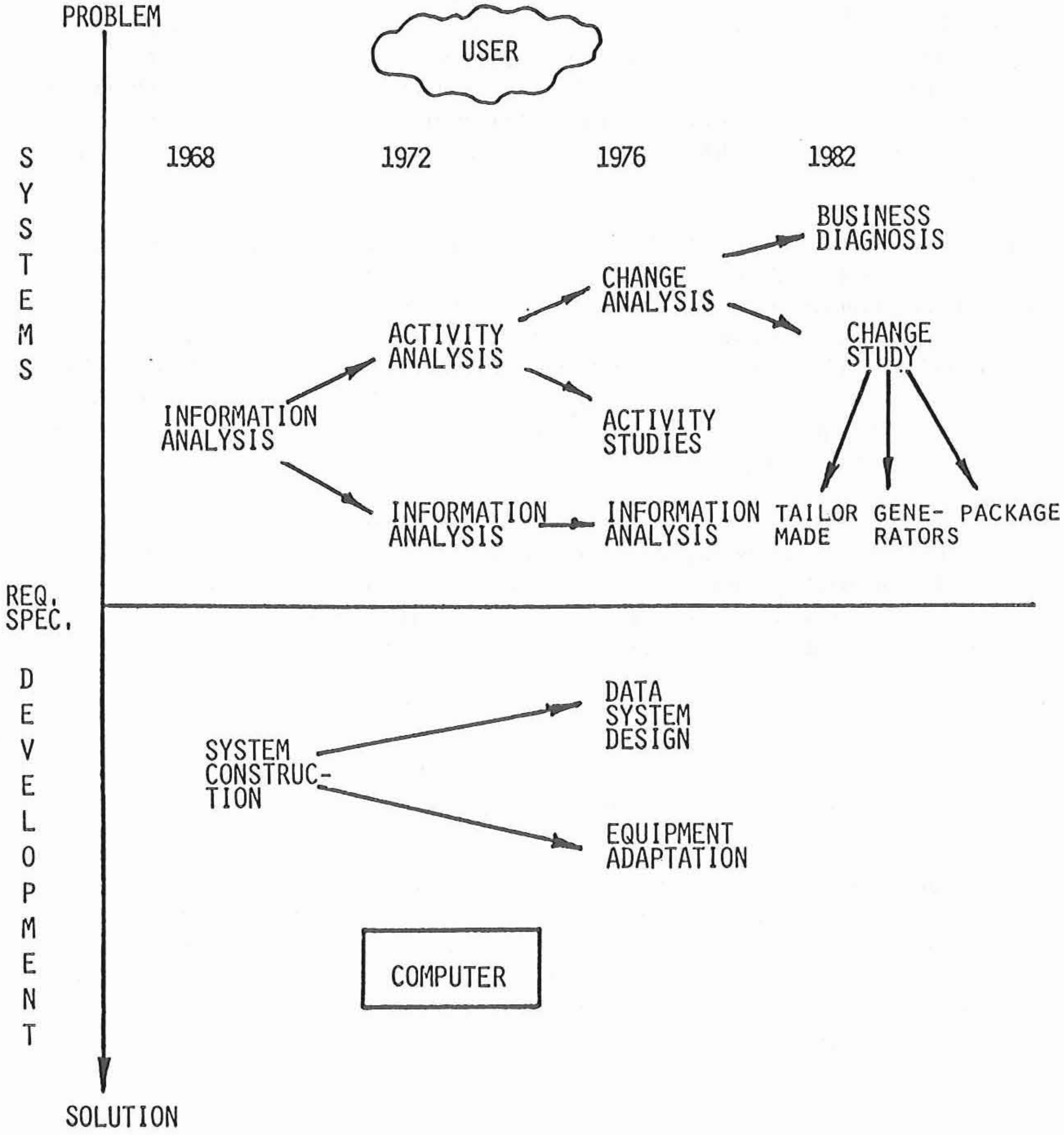


Figure 8 Evolution of the ISAC-approach

The classification also shows a historical evolution of methodologies. The first approaches for development of computer-based information systems were function-oriented. As a reaction to these data-driven approaches somewhat later emerged on the market emphasizing an object view of the reality. They proposed that information systems development should be built on data, which are more stable than functions [4,21]. Both function- and object-oriented methodologies originally proposed a static description of information systems. As a reaction to these behavior-driven approaches emerged considering properties such as triggers, dynamics and time-sequences. We use the term event analysis for this kind of approaches. A special kind of event analysis is the use of routine analysis. An interesting comment is that routine analysis has been used for a very long period of time when designing manual routines in information systems.

At the beginning the three different aspects or "schools" were competing with each other. But nowadays they more and more seek for appropriate combinations. The motive for this is that the three aspects need to be complemented in real life applications. We think that a three-dimensional classification in function, object and event analysis could be a solid base for comparison of methodologies for information systems development. See figure 9.

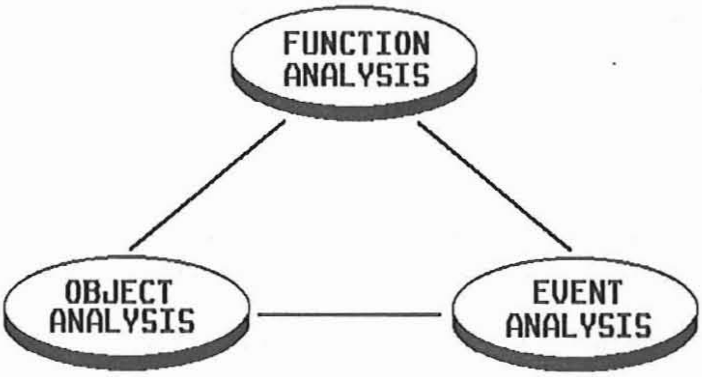


Figure 9 Classification of methodologies

## 6.1 Function Analysis

Function-oriented methodologies emphasize the analysis of purposeful *functions* and their *connections* within a delimited activity area. The development work follows a top-down approach with hierarchical decomposition of functions in a number of levels:

1 Activities and subactivities

2 Processes and subprocesses

3 Programs and program modules

For every function we define appropriate inputs and outputs. It is often recommended to start the analysis with the outputs or the desired results from a function. The connections between the functions is shown by the input/output sets. We can distinguish two major types of connections:

| | | | |
|---|---|---|---|
| 1 | Physical | flow of goods/persons | (operative aspect) |
| 2 | Information | flow of messages | (administrative aspect) |

The function analysis is documented in function graphs, like A-graphs or data flow graphs. All of the function-oriented methodologies handle information flows but a few focus attention on physical flows, foremost Scandinavian approaches. But at the end our interest is to sketch the functions and their information needs. A typical way of working with function analysis is as follows:

1 Break down the business area into purposeful functions

2 Indicate the processing philosophy for the functions (manual, computer-based)

3 Define exactly inputs/outputs for the functions and make when necessary descriptions of the processes

## 6.2 Object Analysis

Object-oriented methodologies emphasize the analysis of *objects* and *information (data)* about them in a delimited activity area. This "school" has many names like object/information/data/conceptual modelling. A typical way of working with object analysis is as follows:

1 Identify purposeful objects (entities) in the business area

2 Describe necessary relations (associations) between two or more objects

3 Indicate appropriate information (data) about the objects and the relations;
this is called attributes (properties) and consist of both identifiers and descriptors

4 Define different constraints for objects, relations and attributes.
Examples are cardinality, transactions and value domains

Object analysis is documented in special graphs, often called data or conceptual models. These could have a variety of complexity depending on how much constraints you want to show in the models. A possibility is to have text pages and tables as complements to the data models.

Data models of a reality can be constructed in different ways. It is often recommended that we first make many small, local user models and then integrate these in several steps to a whole, global data model. This is called view integration of user needs. The data model can be used as a basis for normalization when we want to design data bases for an application.

## 6.3 Event Analysis

Event-oriented methodologies emphasize the analysis of behaviour properties such as dynamics, triggers and transactions in a delimited activity area. The most essential thing is to sketch the *time sequence* for functions and data in a system. A special kind of *event* analysis is the use of routine analysis. More exactly a *routine* (or an errand) triggers a series of events and normally goes through several functions and uses different forms of data, all this in a certain time order. A typical way of working with routine (event) analysis is as follows:

1 Indicate purposeful routines (with events) for the business area

2 Define relevant functions and data sets for every routine;
identify actual interest groups which are responsible for different parts of a routine

3 Describe the time sequence and the behaviour logic of the routine such as triggers and alternative/combined actions

Routine analysis can be documented in different ways, like graphs, matrices and tables. Such routine sketches can be based on either some pedagogical (popular) symbols or some formal (correct) rules. Most methodologies for routine analysis focus on information flows in a system (administrative aspect). Very few approaches also consider the physical flow of goods (operative aspect).

Another way of using event analysis is to specify possible and allowed order between different parts in a system. This is of special interest for dialogue-based systems. In a dialogue-based system we must describe the structure and order between different dialogues (or menues). This can be documented in State Transition Diagrams (STD). A dialogue is often a smaller part of a business routine.

Event analysis is a complement to function and object analysis. Sometimes you tackle limited behaviour aspects also in function analysis (description of processes) and object analysis (description of transactions). But often it is not enough for synchronization of functions and objects/data to each other in a proper manner.

## 6.4 Examples Of Methodologies

We will now give some examples of methodologies for function, object and event analysis. This is a selection of some well-known approaches from Scandinavia and from the international market. Most methodologies have a focus on one aspect and some have an equal emphasis on two aspects. Here are some examples:

| | |
|---|---|
| *Function analysis:* | ISAC/SIV, MBI/SAK<br>HIPO, SADT, SASD (Gane&Sarson), SA/SD (Yourdon) |
| *Object analysis:* | EASYDOC/SYSDOC, OH, SASMO/SCB, SIMOL/CMOL, SUDD<br>EAR, NIAM, REMORA |
| *Event analysis:* | MEKAN, Wall-graphs, SEB-bank, Rutinflöden<br>JSD/JSP, STD, ABN-bank |
| *Function/object-analysis:* | DATA LOGIC (with SA/SD), ROS (with MBI/SAK/SUDD) |
| *Function/event-analysis:* | SIM, SMX |
| *Object/event: analysis:* | OBS, SVEA (with SASMO) |

There are several computer supports emerging on the market today. They support either a specific methodology or a group of different methodologies from at least two of our aspects. We call the latter group integrated tools. Here are som examples:

| | |
|---|---|
| *Function analysis:* | GraphDoc (for ISAC), SIV-DOC, DASAK (for SAK) |
| *Object analysis:* | Modellator (for EASYDOC/SYSDOC and SASMO) |
| *Event analysis:* | RUTH (for Rutinflöden), Speedbuilder (for JSD) |
| *Integrated tools:* | BLUES (with ISAC), DEFT, Excelerator, IEW, RAMATIC |

## 6.5 Transformation

In a real situation we often need to combine at least two of the three aspects function, object and event analysis. We need an appropriate mix of the aspects. Normally one of the aspects have a dominating role, i.e. function analysis when we have a process-complex application. There is a need for more research of how to transform the three aspects to each other. We can on a crude level think of three ways of working with the aspects, namely in:

- Sequential way
- Parallel way
- Iterative way

There is a good motive to start the development work with at least a crude function analysis. This gives a delimitation of the business area as a basis for further discussions on data use and routine handling. A perspective for working with the aspects could therefore be as follows:

1 Function analysis

2 Object and event analysis (in parallel)

with possibilities to do necessary iterations. We have some good experiences from this way of working. A proposed methodology for combining function analysis (step 1-3) and object analysis (step 4-7) could be as follows:

1 Break down the business area in purposeful functions with the help of A-graphs

2 Indicate manual and computer-based functions

3 Define the interfaces (inputs/outputs) between functions

4 Describe relevant interfaces with the help of visual layouts

5 Construct a local data model for every interface

6 Integrate the local models to a global data model

7 Check consistency, completeness and workability

The function analysis is a top-down process while the object analysis is a bottom-up process [10]. A proposed methodology for combining function analysis (step 1-3) and event/routine analysis (step 4-7) could be as follows:

1 Break down the business area in purposeful functions with the help of A-graphs

2 Indicate manual and computer-based functions

3 Define data sets (inputs/outputs) between functions

4 Indicate purposeful routines (with events) for the business area

5 Define which functions and data sets in the A-graphs that are affected by each routine

6 Describe the time sequence for each routine going through the A-graphs

7 Construct a routine sketch or a state transition diagram (STD) for every routine showing the order between different funtions/data sets in the A-graphs

## 6.6 Car Repair Company - an Example

We will now illustrate how we can work with the three aspects function, object and event analysis in an example from a car repair company. The example follows the transformation rules outlined in section 6.5 above. The business area of the car repair company is documented according to:

1 *Function analysis*
A-graphs (ISAC) using GraphDoc (figure 10)

2 *Object analysis*
Data model (EASYDOC) using Modellator (figure 11)

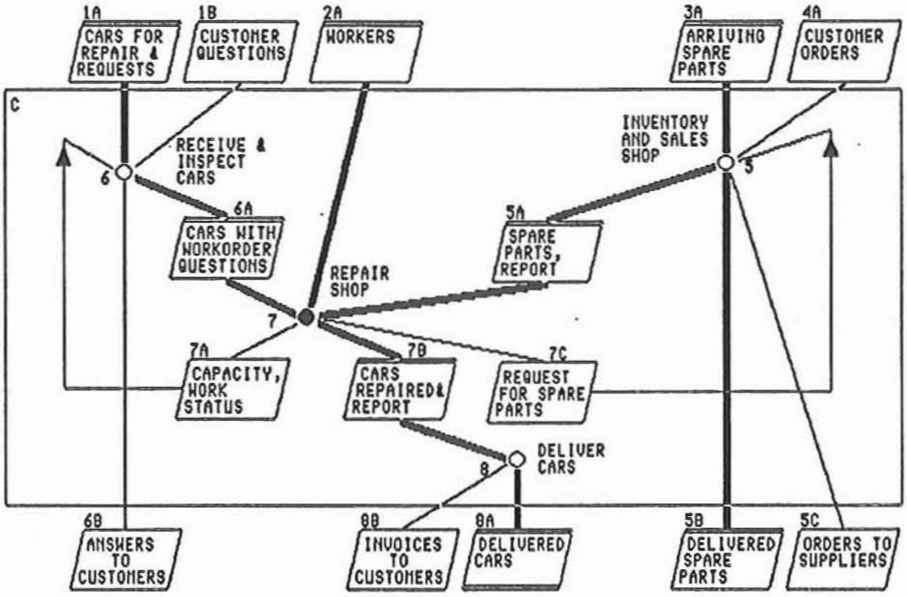
3 *Event analysis*
Routine sketches (Rutinflöden) using RUTH (figure 12a, 12b)

We have documented the current situation of our car repair company. The function analysis consists of two A-graphs; an overall graph of the "car repair company" and a detailed graph of the "repair shop". The object analysis consists of a data model showing a graph of relevant objects and their relations and a text list of attributes; all this in a car repair environment. The event analysis consists of two routine sketches for "car handling" and "customer questions".

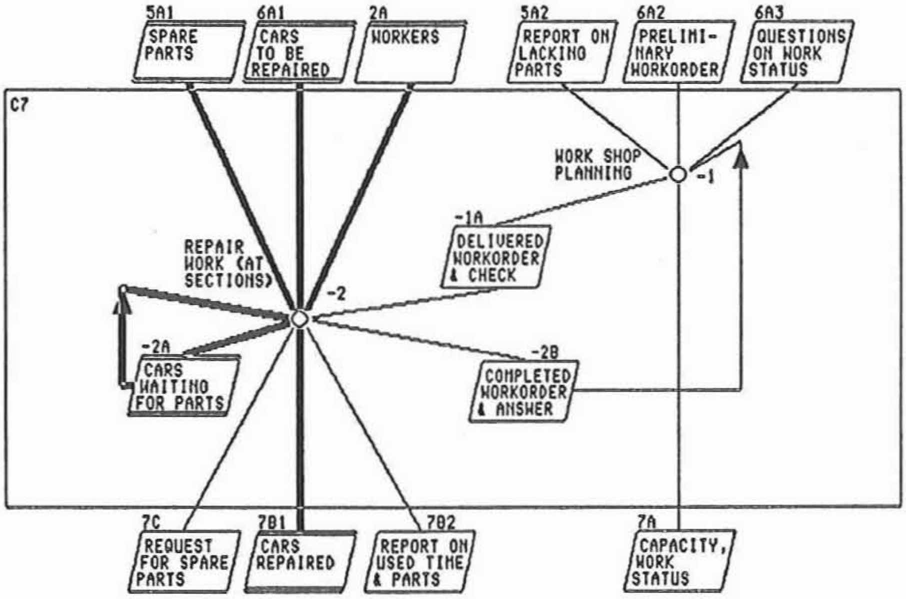
# 7 Conclusions

The purpose of this paper is to classify the process of information systems development according to a frame of reference. The main perspective is that information systems should support business activities. See figure 13. We have here the business activities in focus. There are many factors in a company which contribute and interact in order to achieve desired results in the business activities. On a crude level we have strategies for the physical flow of goods, information systems and organization of work.

Car Repair Company
A-graph CO



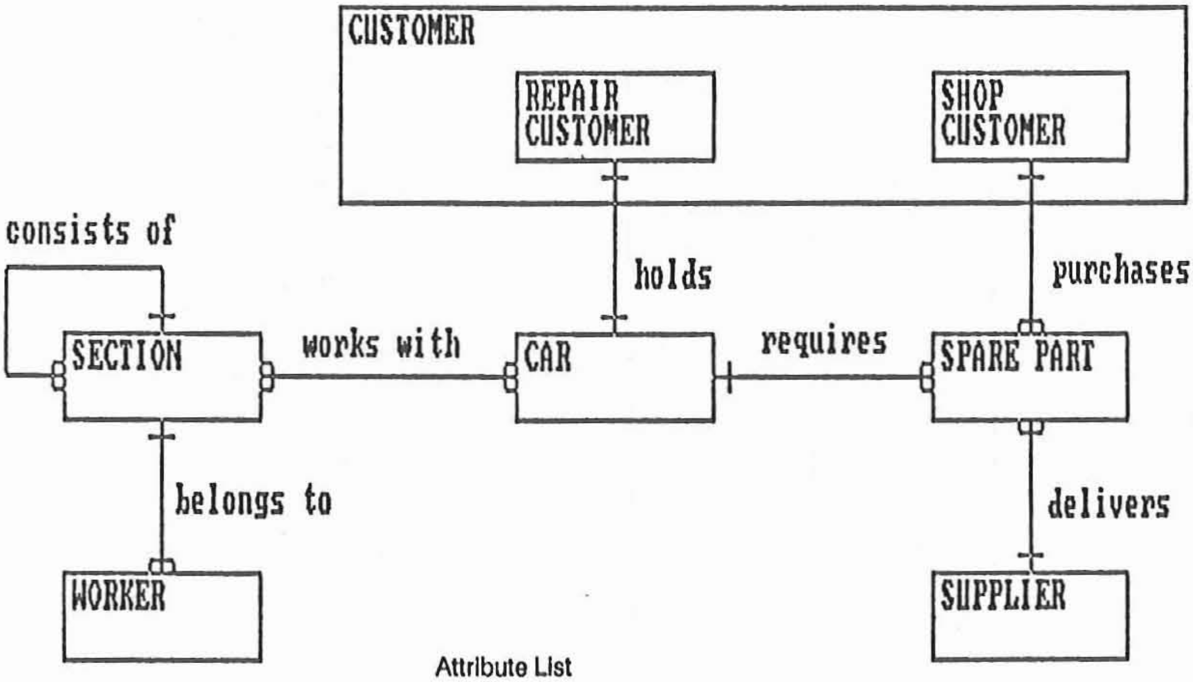
Repair Shop
A-graph C7



5A1 SPARE PARTS
6A1 CARS TO BE REPAIRED
2A WORKERS
5A2 REPORT ON LACKING PARTS
6A2 PRELIMI- NARY WORKORDER
6A3 QUESTIONS ON WORK STATUS
C7
WORK SHOP PLANNING
-1
-1A DELIVERED WORKORDER & CHECK
REPAIR WORK (AT SECTIONS)
-2
-2A CARS WAITING FOR PARTS
-2B COMPLETED WORKORDER & ANSWER
7C REQUEST FOR SPARE PARTS
7B1 CARS REPAIRED
7B2 REPORT ON USED TIME & PARTS
7A CAPACITY, WORK STATUS

**Figure 10 Function analysis (GraphDoc):**
**Car repair company with repair shop**

Car Repair Environment
Data Model



Attribute List

CAR
Car-no
Defect
Workstatus
Work performed
Lead time
Used repair time
Used parts
Cost

SECTION
Section-no
Capacity
Workload

WORKER
Employee-no
Skill

CUSTOMER
Customer name
Adress
Telephone

REPAIR CUSTOMER
Customer name
Address
Telephone

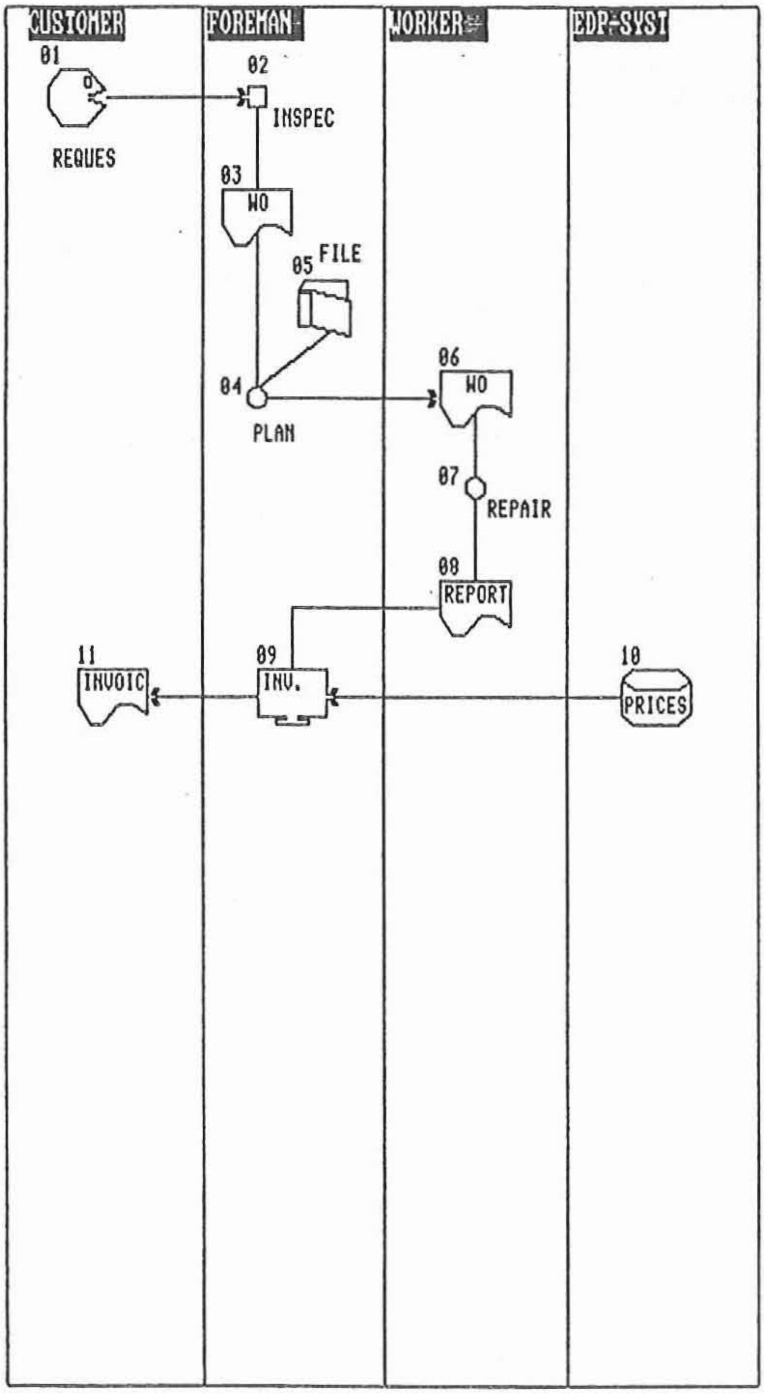
SHOP CUSTOMER
Customer name
Address
Telephone

SPARE PART
Part-no
Number in stock
Part status
Price

SUPPLIER
Supplier-name
Address
Telephone

Figure 11 Object analysis (Modellator): Car repair environment

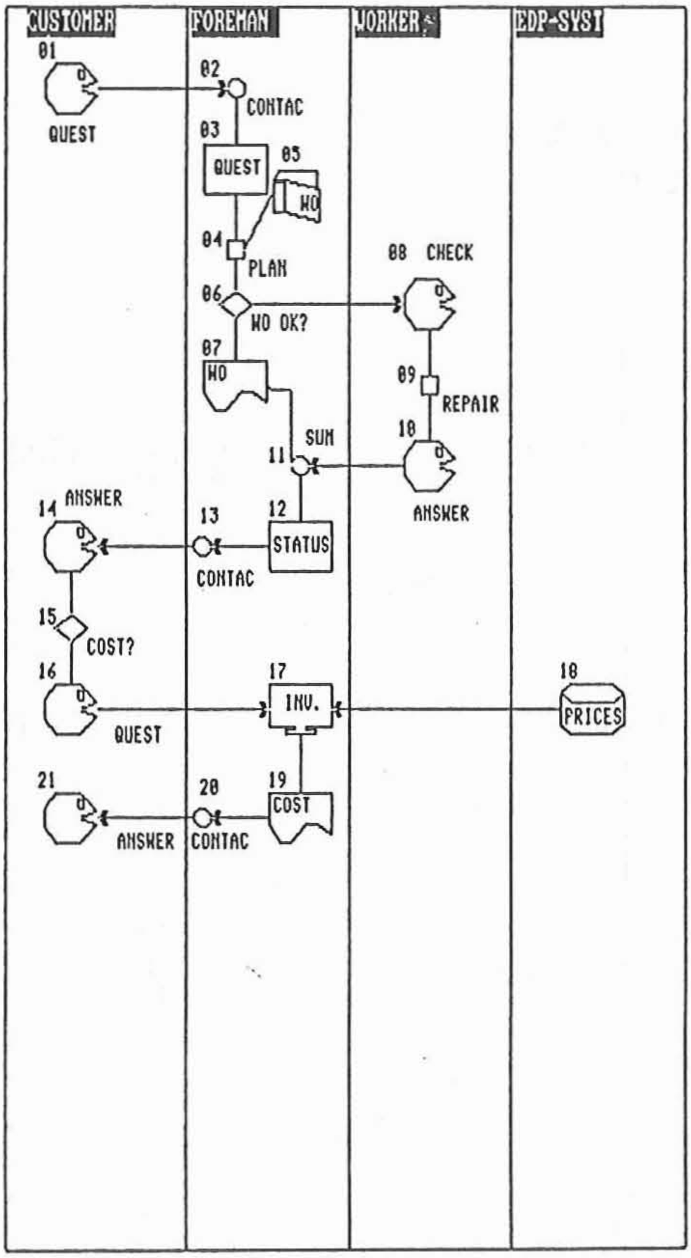
Car Handling
Routine Sketch



- 01 REQUES
Customer requests
- 02 INSPEC
Inspect cars
- 03 WO
Preliminary workorder
- 04 PLAN
Work shop planning
- 05 FILE
File on statistics of repair works
- 06 WO
Delivered work order
- 07 REPAIR
Repair work (at sections)
- 08 REPORT
Report on used time & parts
- 09 INV.
Invoicing
- 10 PRICES
Prices on work time and spare parts
- 11 INVOIC
Invoices to customers

**Figure 12a Event analysis (RUTH):**
**Car handling routine**

Customer Questions
Routine sketch



- 01 QUEST
Customer questions on work status
- 02 CONTAC
Customer contact
- 03 QUEST
Workorder questions
- 04 PLAN
Work shop planning (control of status)
- 05 WO
File on actual workorders
- 06 WO OK?
Can you get info about status from work order?
- 07 WO
Actual workorder
- 08 CHECK
Check on work status from actual workers
- 09 REPAIR
Repair work at actual stations
- 10 ANSWER
Answer about work status from actual workers
- 11 SUM
Work shop planning (summarize of work status)
- 12 STATUS
Work status
- 13 CONTAC
Customer contact
- 14 ANSWER
Answers to customers about work status
- 15 COST?
Is the customer interested in estimated cost?
- 16 QUEST
Customer questions on estimated cost for repair
- 17 INV.
Invoice estimation
- 18 PRICES
Prices on work time and spare parts
- 19 COST
Estimated cost for repair

Figure 12b Event analysis (RUTH): Customer questions routine

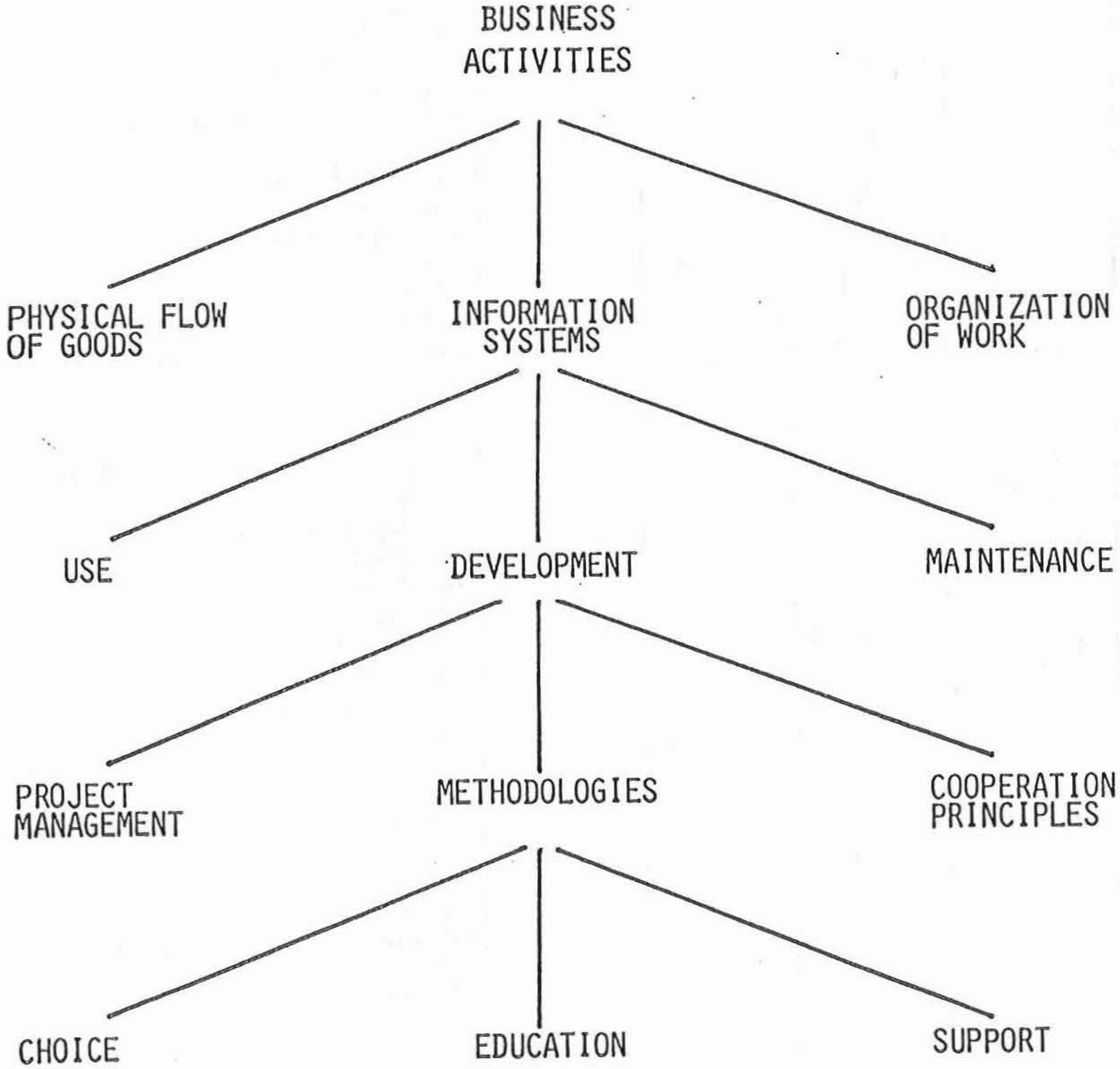


Figure 13 Factors contributing to business activities

Information systems could be of different kinds such as package, generated or tailor-made systems. If information systems should give a good support for business we have to consider the whole life cycle. There is always possibilities to improve development, use and maintenance of information systems. Information systems development is influenced by many factors such as project management, methodologies and cooperation principles. These factors have to be coordinated in a real situation.

Information systems development methodologies can have positive effects in a company. But this requires that the methodologies lead to efficient development, which in turn lead to high-quality information systems, and furthermore to a good support for the business activities. It is also important how a company choose, educate and support a methodology. We have today a huge number of methodologies on the market. But it is rather easy to classify them after function, object and event analysis. In the future a useful methodology must be supported by powerful computerized tools.

# References

[1] ANVESKOG, LENNART & JÄRPERUD, JAN & LUNDEBERG, MATS & MELIN, SIGVARD & NILSSON, ANDERS (1983).
Verksamhetsutveckling - Att Anpassa Standardsystem, Studentlitteratur, Lund (in Swedish, English summary of the SIV-methodology for "Adaptation" is available from Institute V)

[2] ANVESKOG, LENNART & NILSSON, ANDERS & NORD, INGE (1984).
Verksamhetsutveckling - Att Välja Standardsystem, Studentlitteratur, Lund (in Swedish, English summary of the SIV-methodology for "Choice" is available from Institute V)

[3] BOSTRÖM, BERNT & NILSSON, ANDERS & SELLDE'N, JAN (1986).
Systemering med Datorstöd, Esselte Studium, Stockholm (in Swedish, English translation is planned)

[4] BUBENKO JR, JANIS A. (1986).
Information System Methodologies - A Research View, SYSLAB Report No. 40, University of Stockholm (Also published in [19] )

[5] CANNING, RICHARD G. (1979).
The Analysis of User Needs, EDP Analyzer, Vol. 17, No. 1, January 1979

[6] EDMUNDSON, BOB H. & JEFFERY, D. ROSS (1984).
The Impact of Requirements Analysis upon User Satisfaction with Packaged Software, Information & Management No. 7, 1984, pp. 83-90, Elsevier, North-Holland

[7] FITZGERALD, G. & STOKES, N. & WOOD J.R.G. (1985).
Feature Analysis of Contempary Information Systems Methodologies, The Computer Journal, Vol. 28, No. 3, 1985, pp. 223-230

[8] GASPARSKI, WOJCIECH & PSZCZOLOWSKI, TADEUSZ (1983).
Praxiological Studies - Polish Contributions to the Science of Efficient Action, PWN- Polish Scientific Publishers, Warszawa and D. Reidel Publishing Company, Dordrecht, Holland

[9] GROSS, PAMELA HB & GINZBERG, MICHAEL J. (1984).
Barriers to the Adoption of Application Software Packages; Systems. Objectives. Solutions, No. 4, 1984, pp. 211-226, Elsevier, North-Holland

[10] HANANI, MICHAEL Z. & SHOVAL, PERETZ (1986).
A Combined Methodology for Information Systems Analysis and Design Based on ISAC and NIAM, Information Systems, Vol. 11, No. 3, pp. 245-253

[11] KARAGIANNIS, DIMITRIS & SCHNEIDER, HANS-JOCHEN (1986).
Knowledge Based Systems for Information Systems Development - A case study with ISAC-prototypes, Institut für Angewandte Informatik, Technical University of Berlin

[12] LANGEFORS, BÖRJE (1973).
Theoretical Analysis of Information Systems, Auerbach, Philadelphia and Studentlitteratur, Lund

[13] LUNDEBERG, MATS (1976).
Some propositions Concerning Analysis and Design of Informaton Systems, ISAC Specialist Report TRITA-IBADB-4080, Royal Institute of Technology, Stockholm (is available from Institute V)

[14] LUNDEBERG, MATS & GOLDKUHL, GÖRAN & NILSSON, ANDERS (1981).
Information Systems Development - A Systematic Approach, Prentice-Hall, Englewood Cliffs, New Jersey

[15] MALMBORG, ERIK (1984).
Stepwise Formalization of Information Systems Specifications by Extending a Simple Object-oriented Approach, Statistics, Stockholm

[16] OLLE, T. et al.(1984).
Information Systems Design Methodologies - A draft report, IFIP Working Group 8.1 Task Group (unpublished)

[17] OLLE, T. WILLIAM & SOL, HENK G. & COLIN J. (Eds) (1983).
Information Systems Design Methodologies - A Feature Analysis (CRIS II), North-Holland Publishing Company, Amsterdam

[18] OLLE, T. WILLIAM & SOL, HENK G. & VERRIJN STUART, ALEX A. (Eds) (1982).
Information Systems Design Methodologies - A Comparative Review (CRIS I), North-Holland Publishing Company, Amsterdam

[19] OLLE, T. WILLIAM & SOL, HENK G. & VERRIJN STUART, ALEX A. (Eds) (1986).
Information Systems Design Methodologies - Improving the Practice (CRIS III), North-Holland Publishing Company, Amsterdam

[20] STEVENS, WAYNE P. (1985).
Using Data Flow for Application Development, BYTE, June, 1985, pp. 267-276

[21] SUNDGREN, BO (1984).
Conceptual Design of Data Bases and Information Systems, University of Linköping and Statistics, Stockholm

[22] YOURDON, EDWARD (1986).
What ever happened to Structured Analysis?,
Datamation, June 1, 1986, pp. 133-138.