Information Systems Development Supporting Methodologies With Computerized Tools

Bernt T. Boström

EPOC System AB
Box 9105
S-126 09 HÄGERSTEN, Sweden

Abstract: In this paper computerized tools are discussed from a methodological and pragmatic viewport. Development strategies are considered in relation to a suggested classification for tool comparison. A global model describes how three different aspects on information systems development can be supported by a set of tools. Function analysis (GraphDoc), Object analysis (Modellator) and Event analysis (RUTH) are covered with one specifically designed tool for each aspect. The future development of tools is also discussed.

Keywords: CASE tool, computer support, data model, development strategy, function/event/object analysis, Global Model, Information systems development, methodology, routine.

Please note: This paper is related to the report by Anders G. Nilsson "Information Systems Development: A Frame Of Reference And Classifications".

1 Need For Computer Support

Is there really a need for tools when new information systems are developed?
If we regard information systems of a certain complexity, my answer to this question is YES!
However, there are some important factors to consider.

To manage the development of a complex information system, we should use a methodology or a combination of methodologies which will help us. How to select the right methodology is a question of what kind of information system we are about to develop. How to use an appropriate approach will be dealt with later in this paper. Here I want to point out the important role a tool might play in the selection process.

Today a methodology not supported by a tool will have great problems to gain acceptance. This can best be seen from the effort many organizations put into the evaluation of new tools. As there is an enormous number of new tools offered to the market, the analyzing process might become a paralyzing process. There is also another danger. When a tool is evaluated it is too easy to be impressed by the technique of the tool itself and disregard the methodological contribution a good tool should provide.

When a methodology is used without the support of a tool, there will usually occur several examples of "rule-bending". Users without proper tools will invent their own individual modifications in notations, rules and overall use of the selected methodology. This is sometimes referred to as "a high degree of freedom". To avoid rule-bending a suitable methodology-based tool will be very supporting.

The term "tool-bending" refers to the use of a tool which is not methodology-based or based on a different methodology than the one used. The result is often a discrepancy in notation and rules. Tool-bending will occur mainly when an organization is using an existing tool for a purpose it was not designed for. In order to avoid "bending" of any kind one must use a tool which has been designed in close synergy with the methodology used.

Tools normally mean expenses, so what benefits can we expect?
There are several ways in which a tool can simplify and speed up the information systems development. Some of them are listed below.

- *Automatic Consistency*
will eliminate contradictions and related errors.

- *Increased Efficiency*
as tedious repetitive tasks are done automatically with the tool.

- *Increased Acceptance*
to use methodologies as the use of a tool is considered fun.

- *Creativity can be Fully Used*
as a tool will enable a new work technique. With the same time resource it is possible to evaluate more alternatives which should allow for better solutions to a presented problem.

- *No "Rule-Bending"*
as a methodology-based tool should know the rules from the methodology. This will also give the user a good guidance in the methodology.

- *Increased Precision*
because it is easier to manage more complex information system with a tool than one can do by hand.

- *Increased Safety*
as a tool can take care of version numbers to avoid update errors. A tool will only allow authorized people to access the documentation.

2 Development Strategies For Tools

So far we have dealt with the importance of a good tool-to- methodology relation.
What about the tool itself?

First of all we need a way to understand and judge the behaviour of a tool. Tools can be classified in several different ways depending on what perspective we have. To separate the different properties of a tool enables a better, more objective basis for evaluation and comparisons of different alternative tools.

The first way to rank tools is depending on their level of methodological functionality:

A1) Drawing packages where symbols and lines are just symbols and lines.
These tools do not know what they are drawing. There are no methodological rules available inside the tool. Only symbols and notations are covered.
A2) Rule-based tools where some pre-defined methodological rules are built in.
These tools can check the validity and consistency of the documentation within one and the same document.
A3) Rule-based tools as in A2 above but checking is done within a full set of documents.
A4) Generic tools where new rules can be entered and checked by the tool.
Another application for such a tool could be the generation of a graph or a data model from a text input.

The second classification looks at a tool to find how many phases of the total development cycle it covers. This judgement is regardless of how many alternative methodologies the tool covers and must be done in comparison with the development cycle used by an organization.

B1) One step is covered
B2) Some steps are covered
B3) Most steps are covered
B4) All steps are covered.

The third classification estimates the learning threshold required to use the tool. A complicated tool which requires extensive training should for instance be used by an expert.

C1) Easily introduced tools for temporary use by non-experts (less than 1 day training).
C2) Non-complicated tools used by non-experts with some assistance.
C3) Expert tools used by experts and partly by ordinary users.
C4) Expert tools used entirely by experts for central information systems development.

These classified properties can be mapped into a diagram to give a visual image of a tool.

Most of the tools available today support only a limited part of the total information systems development cycle, like the drawing of graphs or generation of code. They are usually related to ONE specific methodology, which they cover more or less. Sometimes they have a user-selectable notation. The information used by the tool is contained in a specifically designed database. These tools are normally bought for the use in a specific project as the price is relatively low. The organisations behind these products are usually small companies often related to the academic world.

A limited number of tools try to support the entire information systems development cycle. These tools have one common data dictionary or encyclopedia used by several separate modules. They sometimes try to cover more than one methodology. To learn such a tool normally takes extensive training and the tool will become an expert's tool. These tools are normally quite expensive and thus considered to be an investment. The buyer has to accept the entire tool concept as their own policy. These products are often developed by large consultant groups.

Today there is an ever growing number of new tools offered to the market. An estimation of the market is 70-100 commercially established products. The expected rate of new products is some 50 per year. We must remember that new methodologies are not developed at the same rate. If the majority of these tools are not based on a good methodology or violates the supported methodology, this development could even be harmful. Concepts named Information Engineering (IE) and Computer Aided Systems Engineering (CASE) are used for some tools. The idea is that information systems can be produced in a more or less standardized way similar to when a CAD system is used for the design of a house. Regarding information systems development the "engineering approach" is rather weak in the early phases and powerful in the data base design and code generation.

The market wants to have tools that are related to a proven chain of methodologies supporting the entire information systems development cycle. Such a development takes a lot of resources. This is also the reason why the majority of tools only covers a limited part of the total information systems development cycle. Later in this paper I have suggested one possible solution to this problem.

It is an obvious trend that information systems will play a strategic role in an organization. This will generate a demand for tools that are easy to use, easy to learn and easy to access. To cope with this situation I find it important that tools should

- *Use a Standard Personal Computer*
  with high-resolution graphics so the tool can be used in as many situations as possible.

- *Be Efficiently Designed*
  not requesting more than standard DOS RAM and not requesting co-processors.

- *Be Easy To Learn*
  not requesting a comprehensive training.

- *Act According To a Methodology*
  to establish a uniform strategy for information systems development.

3 The History of One Methodology-Based Tool

In 1981 I learnt about the ISAC methodology [6] at the University of Stockholm. The first part of that methodology deals with activity graphs. This technique is referred to as SDA (Systematic Description of Activities) [1]. One year later I had the opportunity to use a tool for the production of activity graphs. This prototype tool (VSB) was designed for an HP minicomputer. Although this tool was rather simple, I could feel the useful support a well designed tool could provide.

A market survey in October 1982 indicated a total of nine known tools for the ISAC-methodology. None of them had been installed outside the designing site. Among the designers we found organizations like Volvo, IBM (ISACAID), Roslagsdata (VSB) and the ISAC-group (IA2). These tools were built for mainframes and minicomputers. We must remember that so far the IBM PC had not yet been introduced in the Swedish market.

In december 1982 we established a new company for the design of a PC-based tool named GraphDoc. The first presentation of a prototype for the SDA-technique was made to the ISAC User Group in December 1983. The first release of GraphDoc was announced in March 1985. GraphDoc supports the drawing of A-graphs and I-graphs together with textpages and property tables. [4]

4 Experience from Tooling A Methodology

When GraphDoc was introduced as a tool for the SDA-technique many users were quite enthusiastic. One year later the ISAC User Group claimed that GraphDoc had contributed to a second wave for the methodology. Our experience is based on more than 300 installations from industry, banking and consulting agents. Also universities in Sweden, Holland and Norway have used GraphDoc for the training of systems analysts. Only at the University of Stockholm 250 students have used GraphDoc for a practical case study (Car Repair Company). From this experience we have drawn some important conclusions.

- A tool will not produce any miracles on its own because the tool is never better than the methodology it supports.

- Introduce the methodology and use the description technique by hand before a tool is used. The tool cannot teach the methodological perspective. It can only give guidance. Therefore we have adopted a scheme for beginners where we introduce the methodology in the first day and copy the hand-drawn documents by means of the tool the second day.

- To gain acceptance for a methodology the litterature describing the methodology must be easy to access. Most available litterature was considered too academic to attract users involved in the information systems development. This was the reason for a new Swedish book, which has been translated into Dutch and Norwegian [3].

The advantages mentioned by users are that GraphDoc

- acts according to the methodology which is easy to recognize,
- produces nice documentation which is uniform and easy to read,
- supports the introduction of a methodology,
- provides automatic consistency,
- makes a tedious task more fun,
- allows higher precision.

The disadvantages mentioned by users are that

- GraphDoc has a limited drawing-area of one screen.

This is a trade-off between overview and space. The methodology sets the practical width of the graph. If the height exceeds one screen the sets above and below the graph cannot be seen at the same time.

-An underlying structure cannot be accessed until the superior graph is correct.

This is an example of the rules built into the tool to encourage a true top-down approach suggested by the methodology.

- A tool puts restrictions on the user.

This can be experienced as beeing a drawback but is necessary to obtain a uniform development strategy according to the selected methodology.

- There is a need for a full set of tools which support the entire information systems development chain. This issue is discussed later in this paper.

5 Current Situation

In order to cope with the development of information systems for different situations we have worked on a global model showing the relations between the three aspects for systems development. In the model there are three separate areas (aspects) for Function Analysis, Object Analysis and Event Analysis.

With this global model we can simultaneously describe and explain an information system from a functional (process), an object (data) or an event (routine) view. It is also possible to explain why all problems cannot be visualized in only one of these three areas.

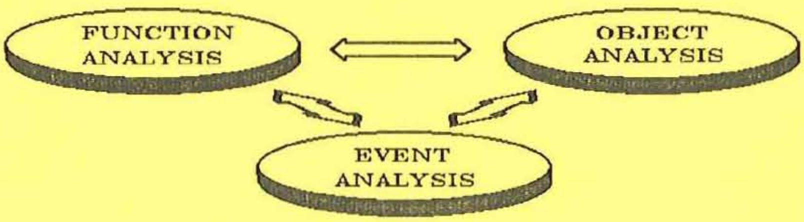


Figure 1 The Global Model View

5.1 How to Use the Global Model

A systems analyst who goes for function analysis will desribe how sets of information and sometimes real sets flow between different activities and processes. This is an excellent technique to show the functionality and the boundary of a system. There can occur difficulties, however, when someone wants to trace the order in which certain tasks are performed as this usually can not be understood from these graphs.

The systems analyst who prefers to use object analysis has several good reasons for it. A data model is normally very stable and will not change much with time. The normal data model, however, often includes events and they are unfortunately not as stable as the data described in the model. The events used in the data model are often procedures or activities.

The third area in the Global Model is something most systems analysts have not dealt with so far. This is a very concrete area used for event analysis, where routines are described. The routine will show triggers and results, time sequence and how different sets or subsets are used by the described processes.

By using all three views in the Global Model it is possible to use each area in its optimal way. When a selected technique tends to be misused, we can go into one of the others.

5.2 Tools for the Global Model

For the functional analysis we are using GraphDoc, which is already an established tool for the SDA-technique in the ISAC methodology.

A new tool Modellator will be used for object analysis with the data models. This product was released for testing in Februrary 1988 and is based upon the ER-approach as recommended by eg. James Martin [7]. A preliminary version of the program has been used at a number of universities with very good results.

Another new tool RUTH is used for the routine analysis. This tool has been developed in parallell with the methodology behind it. RUTH has been used in production since December 1987 for evaluation. When the test period ended in April 1988 more than 600 routines had been described using RUTH.

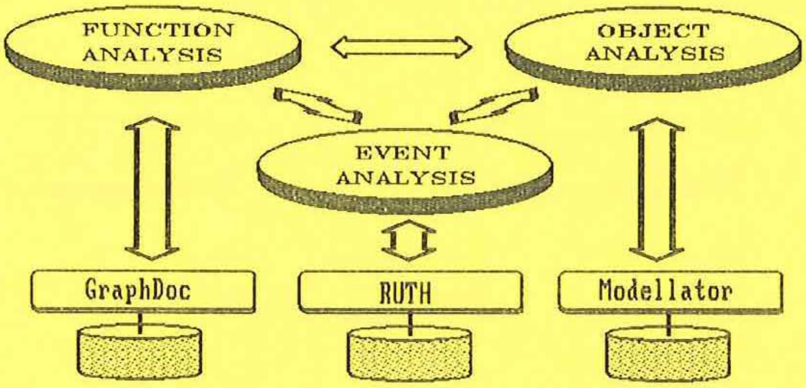


Figure 2 Existing tools for the Global Model

5.3 How to use the right approach?

This is a question of what the task is. In many cases, as these three examples will show, it is obvious whether we should start with one area or another.

My first example is an information system for a mining company, which is a prime candidate for a Function Analysis. Here the SDA-technique has certain advantages as it will manage also physical (real) flows. Then we can show not only the information supporting the activity but also its operative flow. From such a graph we can see what information supports the operative flow ("the vital nerve") directly.

My second example is a lending library system. This is an obvious candidate for Object Analysis as such an activity always will deal with the entities books, borrowers, authors and publishers. By using the Object approach their data model will be stable even if their routines are changed. It is, however, important to exclude events (processes, routines) from the data model as they might change from time to time. A technique for the elimination of events is suggested later in this paper.

My third example is a travel bureau. This is a perfect candidate for Event Analysis as their main concern is to produce service to their customers. This they can do with well designed routines which are easy to adopt to new demands from their customers. Of course the travel bureau is using external information systems, owned by airlines which in turn probably has used a data model for their development.

The three areas in the Global Model can be used in different order. One example is as follows:

1) Perform a coarse Function Analysis to find the system boundary.

2) Generate a coarse Full Data Model based upon step 1.

3) Make a Routine Analysis for the most important routines.

4) Perform a detailed Function Analysis for the activities of interest until the processes in step 3 are reached. Use graph from step 1 as starting point.)

5) Make a small Model for each process (screen) in step 4.

6) Merge all small Data Models from step 5 into the Full Data Model from step 2.

With this combined approach we can use each methodology and tool in an optimal way well within their respective limitations. With a set of well defined rules the transformation process between the steps could be supported by a computerized tool. To be able to define rules at all we to sort out the discrepancies different approaches might create. The problem is illustrated in Figure 3-5.

Figure 3 Functional analysis (activity graph)

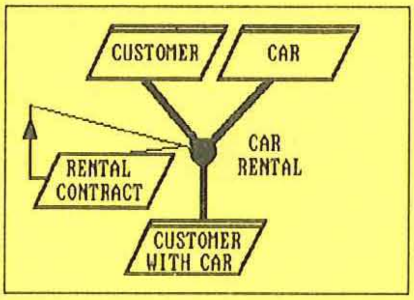
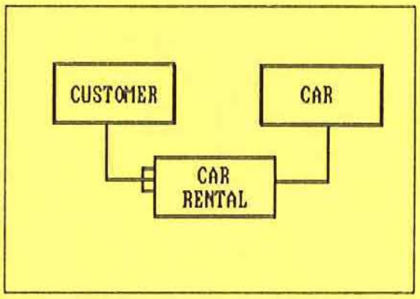


Figure 4 Object analysis (data model)



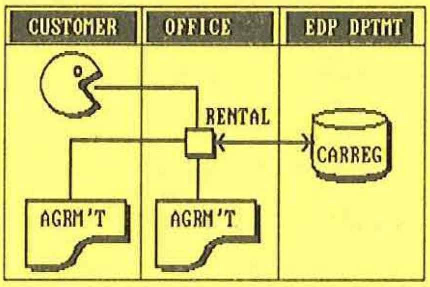


Figure 5 Event analysis (routine graph)

Regardless of selected approach we are analyzing the same business activity or a part of it. If we study the simplified graphs we can see that the activity graph has one activity and four set symbols. The activity CAR RENTAL needs CARS and CUSTOMERS to be at service. This is also a physical flow. The activity also produces RENTAL CONTRACTS which are used within the same activity. However, we cannot see how the rental is managed or what is triggering this event.

The data model contains three entities. Two of them can be related to sets in the activity graph. The entity CAR RENTAL corresponds to an activity. Most data models contain events. This is necessary if the entire analysis is performed with data models only. With our combined approach it should be possible to transfer events into activities or routines. In this example we know a car rental event has happened every time a new RENTAL CONTRACT is generated. If we use the entity RENTAL CONTRACT the data model will contain only real entities. This will improve the stability of the data model.

The routine graph has three columns. We can see the trigger coming in from the top (CUSTOMERS REQUEST). The result of the RENTAL event is two copies of an AGREEMENT of which one copy is forwarded to the CUSTOMER and one copy is kept at the OFFICE.

A data storage CARREG kept at the EDP DEPARTMENT is read and updated during the RENTAL process. From the graph we can understand that the document called AGREEMENT corresponds to the RENTAL CONTRACT in the two other graphs. Still we cannot see any CARS ,Only information about CARS (CARREG) as the routine CAR RENTAL only deals with the paper work. This routine corresponds to an information system inside the activity CAR RENTAL in the activity graph.

When the discrepancies are sorted out the three graphs are more easy to compare. The graphs can be used for validation in comparison with each other. Of course such a comparison should be done with a computerized tool. Therefore we must find a natural way to minimize discrepancies as indicated here.

Figure 6 Function analysis (activity graph)

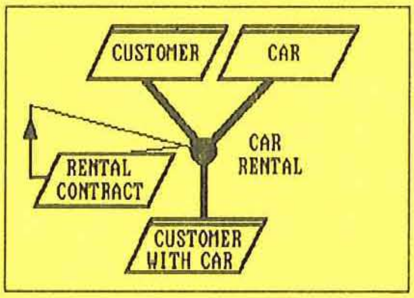
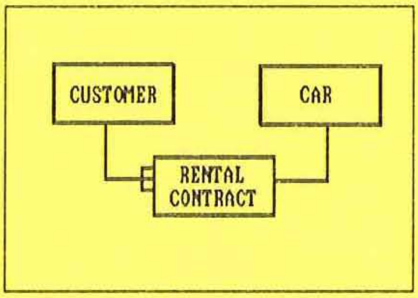


Figure 7 Object analysis (data model)



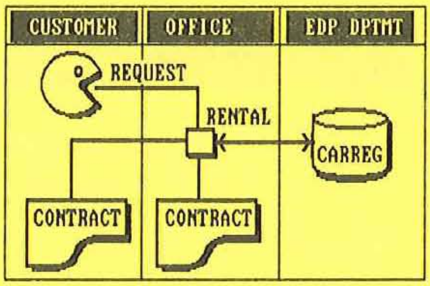


Figure 8 Event analysis (routine graph)

6 Future Development of Tools

As I have mentioned earlier new tools are offered to the market at a rate which far exceeds the development of methodologies. This is a dangerous situation as tool developers might find it more interesting to produce tools that are fancy rather than based on a methodology. There are several examples of tools which have deteriorated during the implementation process.

6.1 Can the AI technique provide a new generation of tools?

Artificial Intelligence has generated great expectations on the development of expert systems. With an AI-based expert system we usually mean a program which reproduces a human experts ability to solve problems or to give advice. Expert systems are based upon symbolic knowledge and useful systems were early designed for medical diagnosis, technical fault-detection and oil prospecting.

Even if there are good examples of useful expert systems, we must remember these application areas were selected because they were suitable for the AI-technique. Expert systems are suitable when decisions can be based on conclusions drawn from a combination of facts and rules. All practical applications are implementations of concrete routines. Information systems development, however, is abstract and therefore not straight forward enough.

Also in the future tools will serve the systems analyst much in the way as a word processor serves an author. The tools will have difficulties in drawing conclusions simply because we have too much of a problem to express the underlying facts. It is exciting though to think about a tool where we just describe the problem vaguely and the perfect solution is typed out on the printer. Until then we have to expect evolution instead of revolution.

6.2 What do we do then?

The Global Model which has been presented in this paper explains how different views on information systems development can be related to each other. There are, however, still some work to do such as

- transformation between different areas of the Global Model,
- development of remaining tools for the Global Model,
- improve the understanding for a combined approach where all three areas of the Global Model are used in synergy.

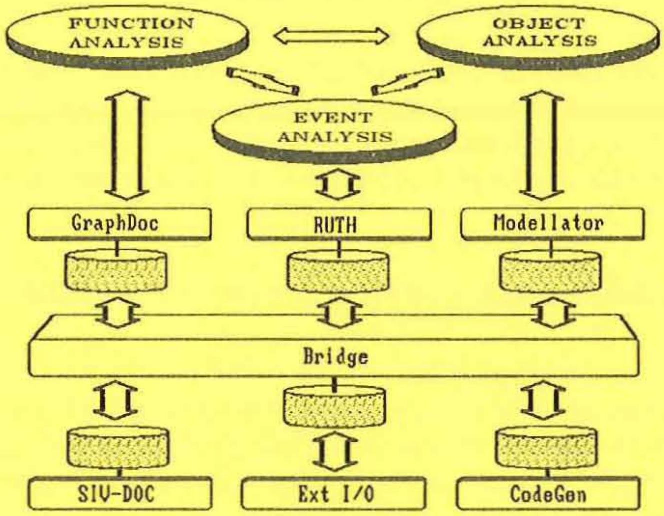


Figure 9 Full content of the Global Model

Most integrated tools are based upon one common data dictionary that holds all information stored about the information system. In order not to slow down the development of our specialized tools, each tool in the Global Model instead has its own database. To obtain integration between the different views a database from one area of the Global Model has to be transformed into another. Transformation between different areas of the Global Model has to be further analyzed . Some work in this area has been done by others [5].

Some parts of the Global Model has not yet been covered with tools. The specifications for the "Bridge", "External I/O" and the "Code Generator" has to be drawn. It is also essential that all tools supporting the Global Model have a standardized user interface.

The tool SIV-DOC is today a prototype related to the SIV-model supporting the choice of standard packages [2].

An important issue is to inform about how the Global Model can be used in the information systems development. In this process the interaction between the areas in the Global Model can be illustrated by this view.

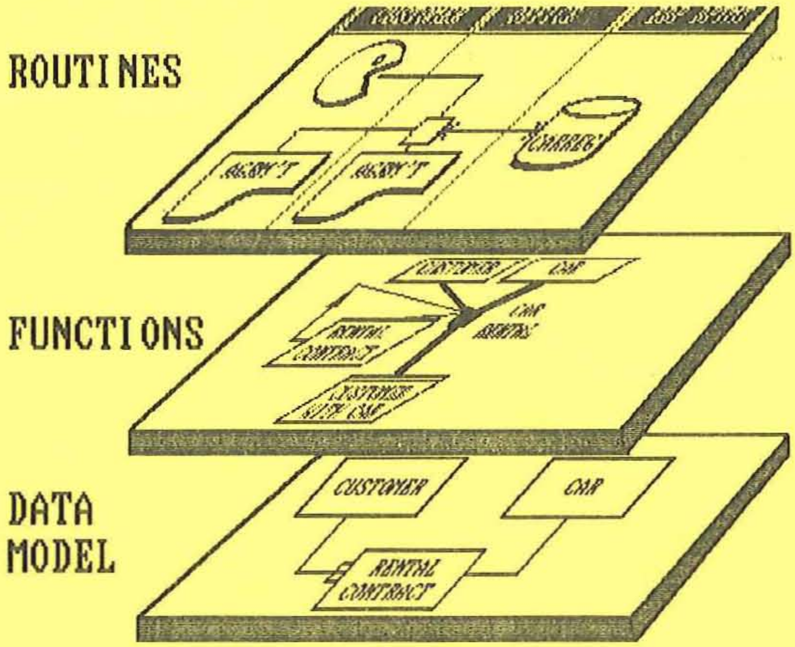


Figure 10 Global Model interaction

The bottom floor consists of the data model with entities, attributes and relations. On the first floor we have the functions with activities and processes, which operate on the entities, their attributes and relations. The top floor contains the routines, which are the concrete descriptions of how tasks are performed.

Object analysis and data driven design has a major benefit as it is based upon a stable data model. In order to ensure this stability we have to eliminate the events from the data model. It is not necessary to restrict the use of events in the early phase of the data modelling. When the data model has been established, however, all entities should be checked for events. When an event is found (for example Car Rental), we can ask ourselves "How do we know this event has happened?". The answer can then be used as the real entity (Rental Agreement).

Sometimes events in the data model have no real entities. Such events we call "fictituos entities". I do not consider them relevant for a data model as they are true events without substantial verification. An example could be the event "accident" related to cars, drivers and passengers, but no accident report (entity) is written.

7 CONCLUSIONS

The purpose of this paper is to increase the understanding for computerized tools supporting methodologies. Therefore I have compiled the information obtained so far.

Tools have until now been more or less copying the way we used to draw with paper and pencil. Computerized tools have a much higher potential. To achieve new levels we must consider how different approaches of information systems development can interact. Therefore I have suggested the Global Model using several specifically designed tools supporting one methodology each.

Use of the Global Model offers a combined approach where we do not have to defend one approach instead of another. It is instead possible to use function analysis, object analysis and routine analysis in synergy.

REFERENCES

[1] ANDERSEN, ERLING & NISSEN, HANS-ERIK
Systematic Description of Activities
(Print available from Insitute V, Stockholm)

[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 & SELLDÉN, JAN (1986)
Systemering med Datorstöd, Esselte Studium, Stockholm (Translated into Dutch and Norwegian. English translation is planned)

[4] GRAPHDOC USER MANUAL,
Epoc System, Stockholm

[5] 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

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

[7] MARTIN, JAMES (1987)
Recommended Diagramming Standards for Analysists and Programmers, Prentice Hall.

[8] 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.

[9] ÖREN, OLE (1985)
Datamodellering, en införing, NKS