A framework for use and classification of CASE–tools in systems analysis and a strategy for implementation

Jan Hernbäck
Ingemar Lindqvist
Hernbäck & Lindqvist konsult AB
Stockholm, Sweden

In this paper we will describe a pattern for the different tasks to be performed during the development of a computer-supported information system. We will concentrate on what should be done during the "business-oriented" phase of the development process, i.e. before detailed technical design begins.

We will describe the main system development tasks in the pattern. We will also describe how a solid foundation for the more detailed work can be laid by using well-defined user-dominated seminars as a met
hod for early, important activities in the development process. The detailed analysis work tasks will be briefly described.

Finally we will use the pattern to describe the scope of, and thereby classify, different CASE-tools. We will also discuss how such CASE-tools best should be introduced in an organization when using our pattern and our methods of work.

UNDERLYING IDEAS

Our pattern and its content are built upon a few basic ideas:

1) The non-technical, business-oriented part of systems development (called Business Area Analysis and Business System Design by Information Engineering disciples, Requirements Analysis by others)
- should not aim at creating a Requirements Specification for future systems design,
- but at specifying a complete, "logical" system describing how everyday business activities will be performed by people and computers.

2) Such a complete, logical system specification should contain different descriptions of the business at hand and logical links between the elements of these different descriptions.

3) Any business is too complex to be fully described by using only one type of description technique. A business could be regarded as having a number of different dimensions, each needing its own description techniques. The number of possible dimensions is infinite, but a limited number must be chosen for practical reasons.

4) We get a reasonably complete specification with a reasonably limited effort by focusing on three dimensions:
* The data dimension, where we describe what the business needs information about and the details of this necessary information.
* The function/process dimension, where we describe into what logical "processing units" the business could be split up.
* The event/procedure dimension, where we describe the different kinds of business events we must handle, and the business procedures needed to handle them properly.

5) If you want to successfully computerize a certain business you must understand the business thoroughly first. This is true for "normal" applications. It is equally true for system development - our own "business". Therefore: To successfully choose and introduce CASE technology you must first decide how system development work is to be performed and what results it should yield.

Real life application

The pattern and methods we present have been in use for some time in a number of Swedish organizations. You find them, e. g., in the four largest Swedish bank groups. In all these banks the pattern has been used for CASE technology evaluation and introduction. It is thus a framework that has proven to be applicable in the real-life systems development of large EDP users.

ANALYSIS OF STRUCTURES

In general

As stated earlier we recommend that the start of the project should be seminar oriented. During a relatively short and intensive period of hard work we put all project members through a series of seminars. These are oriented towards the fundamental issues of analysis; what functions, what data and what business procedures.

Since the objectives of this phase are to establish foundations for the analysis and put the project "on rail" we work with high levels of generalization. We want to create a high-level realistic and stable functional structure. We want to establish the basic entity-relationship structure and we want to establish the main business procedures and the principles of their solutions.

The seminars are NOT activities to unravel the eternal truths, but negotiations among different knowledgeable people about what "truths" should govern this project. Furthermore, we know that these three structures do not capture all the main issues of scope, level of ambition and principal solutions. These seminars are to a very high degree organized in three dimensions to provoke debate on all issues of central importance. A very important result of the seminars are the discussions in themselves and their results in form of decisions or instructions to investigate or negotiate central issues further.

We use the seminar phase of the analysis to obtain a real understanding of what work is to be done. Only when we have achieved that is it meaningful to introduce methods and CASE-tools which cover issues we all by then have come to recognize as important.

Our experience is that by taking things in that order you get, in a very short time, a real understanding of what the analysis work is all about and how it is done. Since all the relevant persons work together all the time they will also have a common base of understanding and will support each other in the learning process.

Together with the project leader and "main user" we establish an understanding of this phase of the project and plan the principal steps including what seminars in which order.

For each seminar we have
- a preparation phase
- the seminar itself
- a documentation and decision phase
- a phase of transition from seminar oriented work to normal project activities

The preparation phase

When do we do it - the planning of each seminar is coordinated with the other seminars. The individual seminar should normally be concluded within one to two weeks.

What people in what roles - from the "eagle-oriented", very knowledgable end user who "makes things happen" to other end users which form a solid and detailed "knowledgebase" Often system analysists and technicians participate, in a more passive role, to become "indoctrinated" during the seminars: their future work is to be governed by them.

We also have the seminar functionaries - the seminar leader (in larger seminars two leaders), well trained and experienced both in the method and in seminar techniques, oriented towards producing results and with natural authority enough to guide and control the seminar.

We have two persons responsible for documentation, one end user
and one systems analyst. Their responsibility is to note down everything of importance: graphics and definitions, decisions taken, ambiguities, short summaries of debates and important differences of opinions.

Where - the seminar should take place well away from the ordinary working day environment in order to ensure no intrusions and enable a total concentration on the business at hand.

The disposition of the seminar - what activites, in what order and who is responsible - has to be determined.
The disposition should include a seminar introduction where we establish:

- what we want to achieve
- how the results are going to be used
- the total seminar plan
- how this seminar is structured

and present the seminar and its members and functionaries

The method used in the seminar - has to be a graphic method. The issues under debate must be caught on the fly. The best way to do this is to enter them into a continuosly evolving graph.

It has to be relatively simple, otherwise people will have a too long introduction period and will feel insecure in terms of what really is said and done. It must create a good basis for the future analysis; it has to be CASE-supported since the seminars only are the starting point. The results will live, change and expand over a longer period of time involving a great number of people

The CASE-tool - part of choosing method is choosing a CASE-tool. The different methods chosen should be supported by and integrated in the CASE-tool. Normally the CASE-tool is not introduced in the seminars but the results are stored there as a starting point for future detailed analysis and for the effective administration of integrated seminar results.

Documentation - the results of the seminar should take the form of a report, containing graphs and definitions and reflecting decisions taken, directives to further analysis and short summaries of important debates.

The seminar phase

*DATA ANALYSIS SEMINAR

Objectives:

The results we aim at are
- a normal Entity-Relationship graph covering the fundamental entities (or, more correctly, entity types) of the business and important relationships between them. Typically such a graph contains between 10 and 15 entities. (Later during the Analysis phase less important entities will be added and the final normalized model will stepwise be built on this foundation)
- definitions of all entities and those relationships whose meaning is not obvious. All major concepts concerning entities are coordinated.
- documentation of primary keys for the entities, including primary keys that need redefinition (if there are good reasons to change the way the account numbers of a bank are built - and there sometimes are - such a change needs careful and time-consuming preparation)
- documentation of ambiguities and differences of opinion that need special examination and negotiation

Seminar program:

The seminar starts with a short introduction during which
- the project leader and/or the seminar leader explains what the seminar is about and why it is important
- we agree upon scope, objectives and principal changes in the business concerning the planned systems development. These are normally already formulated during earlier stages, but are quoted here to make all participants work in the same direction
- the seminar leader holds a "mini-lecture" in how an E-R model is built and how it is interpreted (less than half an hour.
  Only the most fundamental points are raised, not-so-important issues are explained as they appear during the seminar).

After the introduction a list is made of all "things that we need information about". Since the participants normally have little formal understanding yet of what an entity is, the list will contain items that will not appear as entities in the model ("payment date", "overdue notice" etc). This is deliberately accepted - later the seminar leader will gently make the participants understand that some of the "things" are not entities, but rather attributes or different forms of output from the stored information. At this stage it is important not to embarrass the participants, and often there lurks an entity behind the formally incorrect expressions.
If the list tends to become polluted by too many non-entity "things" the seminar leader makes a break to hold a short lecture on e.g. the difference between an entity and an attribute.

The list is ended when the participants start to run out of imagination. This normally takes between one and two hours. A typical list contains some 30 or 40 "things". It normally covers all the resulting main entities in one way or another, but it is not crucial that it is "complete".

We then start building the E-R graph on the wyteboard. Many business areas have at their heart some kind of "business agreement" ("order", "deal" etc) with a "customer". We often build our E-R model by following the life cycle of this "business agreement" - what entities must exist in advance, what entities and relationships appear during the negotiating phase (if there is one), what additions are made when the agreement is made etc. As we add entities and relationships to the model we erase the corresponding "things" from our list. At this stage we also discuss the non-entity "things" to see if we can agree upon that they in fact e. g. describe a property of one of our entities rather than being entities in their own right.
The building phase ends when all "things" on our list have found their proper place in the model and the whole basic life cycle (if there is one) has been dealt with.

When the model is finished we check it by answering a few "test questions", such as:
* Describe some possible changes in the business environment (due to changes in legislation, new technology, new competition etc). Would the model change? Can we make it less vulnerable to change, e. g. by changing the level of abstraction?

At the end of the seminar the seminar leader explains what result the participants should expect in the near future, and when they should expect it.

The seminar is followed by a documentation and decision phase, as described below.

*FUNCTION ANALYSIS SEMINAR

While the E-R model is used to lay the foundations of the data analysis and to define the "business language" to be used during e. g. dialogue design, the function model (in the form of a Data Flow Diagram) is used
- to define the scope for the systems development (i. e. the boundaries of the relevant business area)
- as a structure for the subsequent documentation of details on the business processes
- for the definition of separate analysis sub-areas to be distributed among the members of the project team.

The function analysis seminar is similar to the data analysis seminar in its main structure. In the function analysis seminar, however, we do not list "things that we need information about", but "things we do in our business" and "those that we send information to or recieve information from" (systems, other business areas and other companies alike).
An amazing thing we have found is that the list of "things we do" tends to contain between 20 and 40 items regardless of the size of the business area. It seems that people place themselves on a suitable level inuitively.

When the lists are finished we start clustering "things we do" into "natural" functions ("things" that have more in common with each other than with the environment). The senders/receivers of information are clustered in a similar fashion, if needed, into first level external agents.

The functions, external agents and their connections are then illustrated in a Data Flow Diagram.

The discussions around functions always bring new ideas, principles and possible solutions into daylight. These are heatedly debated, and new, exciting insights, shared by all participants, are created. This is in fact one of the most important results of our seminar method.

During the seminar ambiguities to be solved and investigations to be made are listed when they appear, just like during the E-R seminar.

The seminar is followed by a documentation and decision phase, just like the other two seminars.

*EVENT/PROCEDURE ANALYSIS SEMINAR

The E-R model and the function model together give a good "static" overview of our business area. They help us with important definitions, and they form a solid foundation for the subsequent analysis work. There are however some important questions concerning the "dynamics" of normal business activities that they do not answer. Examples of such questions are:

- Will our business look efficient from the customer's point of view? How long time will elapse from the moment when he expresses his wishes until he gets what he wants (or, possibly, a polite rejection)?
- How much does it cost (manual and computerized activities together) to handle an application for a loan in our bank (or to handle an application for the registration of a new incorporated company, if our "business" is a government authority)?
- Does the way we plan to run our everyday business adhere to corporate security policies, to the relevant legislation etc?

Since our main objective is not just a running computerized system but a smoothly working business we need satisfactory answers to such questions. We therefore work with a third "business dimension" beside the "function" and "data" dimensions, namely the business procedure dimension. A business procedure we define as a description of what is done, and in what order, from the moment a business event occurs (normally an external event, such as a customer applying for a bank loan) until all direct consequences in our business of this event have been taken care of (in our example: the loan is registered in our computer files, the relevant contracts are signed and the money payed out).

The seminar:

The event/procedure seminar is normally preceeded by the two previously described seminars. It is not quite as formally conducted as the other two, but nevertheless important.

The detailed descriptions of our business procedures are made later during the analysis phase. At this stage, during our event/procedure seminar we

- identify the necessary business procedures
- decide upon what objectives should be met during the later procedure definition. Such objectives might include:
  "all customer requests shall lead to a written response within x days"
  "security recommendation "y" must be adhered to in all procedures involving monetary transactions"
- discuss possible changes in our business procedure principles due to new technology, new legislation and planned principal changes discussed in the other seminars
- describe one important business procedure. This is done to give all project members a common feeling for how the objectives should be met and a common understanding of the level and description method to be applied during the later procedure definition work.

To identify our procedures we start by examining our business environment, i. e. the "external agents" in our function model. What business events can each "agent" cause? Together they form an "external event list". Some of the listed types of event are possibly dealt with by our business in a similar way. The list is, therefore, gradually reduced to a list of "business procedures needed to take care of external events".

A similar process follows to identify "internal" procedures: balancing of the books, e. g., is not a direct consequence of any external business event, but must be dealt with in an orderly fashion according to an identified procedure.

From the procedure list one representative procedure is chosen. This procedure is then built. During the building process we especially concentrate on the way our procedure objectives influence the solution.

Later, when we build the rest of our procedures, we work in smaller "joint procedure definition" groups. We first sketch the procedures graphically on the wyteboard, then - when we have agreed upon the basic sequence of activities within the procedure - enter them into a computer-based tool (important notice: since the procedures use the data in our "E-R model" and the detailed processes found on lower levels of our "function model" we maintain formal links between our procedure documentation and the documentation of the two other dimensions).

Phase of documentation and transition to detailed analysis

Even if we have very diligent documentators they can only jot down "short-hand" notes on the most important issues. There will always be much more knowledge created and systematized informally and internally in all the partcipants, including the documentatators themselves.

It is therefore of utmost importance that the documentation and decision phase is very short. Within a day or two after the last seminar of a certin kind the project leader, the seminar leaders and the documentators must meet and compare notes.

To begin with they informally evaluate the seminars:
- have they covered they application area, are we really finished
- have they taken care of every known structural issue
- have people understood what we have done and are they comfortable with the result, including the method used
- do we agree on the results including ambiguities and directives for further investigations

After that the documentation work is planned in detail including work distribution and time alotted based on the planned content of the seminar report and the result of the seminar.

The report must be distributed within approximately a week among the participants including a summons to a decision meeting. It should take place within a couple of days from the recieving of the report.

At the decision meeting a formal presentation of the results is done by the documentatators. They present the graphs by presenting the contents and simultanously drawing them on a wyteboard. Definitions, decisions and directives to further investigations are presented and discussed. In the end a formal decision is taken on the report, a protocol is written by the documentators and included in the report.

It is necessary to have a speedy and smooth transition from the seminar oriented work to ordinary analysis.

We usually have

- created a feeling of success and that "things have started to happen" which it is essential not to lose. People now have expectations which we must meet.
- people who have started a learning process. Project members understand what they are expected to do. They have a good, three-dimensional view of the application etc
  BUT they have only started, we may lose it all if we delay.

And, of course, time is often essential

As soon as possible we want to have all members of the project at work, systematically, effectively and within an organized framework of priorities.

We therefore plan a project meeting in which the transformation of the seminar report into project tasks spread among project members is done.

Before that meeting the project leader has made the relevant project plans. This planning activity is important, and sometimes time-consuming. It constitutes the link between the seminar phase and the ordinary project work.

DETAILED ANALYSIS

Detailed data analysis

In the next step in data analysis we substitute all entities with the data items representing all the information we want concerning the entities.

Every data item must belong to an entity. If we encounter data items which cannot be placed in the entity model we have to revise it so that they can. In practice, this means that every time we find an item we must be able to define its primary identifier among the identifiers found in the entity model.

The data items belonging to an entity are grouped according to a given set of rules, normalisation rules, and the data item groups are illustrated graphically in the same way as the entity model. The graph looks just the same, the difference being that the symbols now represent data item groups and relationships between data item groups.

Data items are normally constructed within the detailed analysis of functions where you decide how the information is processed.
Another source of data items are the existing files; these are, however, subjected to careful analysis to ensure their quality.

The construction of data items is a very critical point in the analysis and has great influence on the quality of the system. It is often practical to have a project data administrator who reviews and coordinates data items, data item groups and the entity model.

The resulting model containing data item groups and their relationships should, in the database construction work, be realized in a data base management system with as little structural change as possible due to optimization.

Detailed function analysis

Functions given in the function model are broken down hierarchically to the level where we can, with complete control, define the detailed logic. This final stage in function analysis results in process definitions which can be programmed manually or by generation.

In order to help us to break down functions we start with analysing the inherent character of the function we are about to break down. The functions (at different levels) can be:

* determined by a business procedure, i. e. the solution of the business procedure determines what processes we must have and what they will contain. For instance, business procedures often contain a computer dialogs which determine the supporting computerized processes.
In this case we first structure the business procedure and the dialogs. Then we form the necessary process structure and define the processes.

* determined by the "logical problem" to solve. Some functions represent a "logical problem", for instance calculating interest on a deposit account. Even if the resulting interest is presented in a business procedure the dialog does not determine the processes. They are determined by, given the basic policies in interest accounting, how we fabricate the algorithm so that we get the result we wish.

In this case we use the internal structure of information dependency to form processes. For instance, in order to calculate compound interest we must know the interest rate and the period, in order to know the period we must establish dates and so on.

* determined by the data structure: the events in a life cycle of an entity determine the necessary processes. This is often very evident for information that does not emerge as a central part of the operative business but rather is used as support in it, e.g. costumer information, information on bonds in a bond dealer system.

In this case we have to study how the entity can emerge in our organisation, for instance a new bond in the market. Then we must follow the different events concerning the entity itself, for instance changes in rates or other conditions until the entity, the bond, expires. All along we have to form processes handling all the events relevant to our organisation.

This analysis is often meaningful to do for all entities in order to check that we take care of all possible situations concerning the entity in question.

Finally we reach a level where we can make a detailed and complete description of the logic of each part. We usually use some form of pseudocode to describe it. A pseudocode is a formalized subset of a language where some specific words are used according to certain rules. There exists a great number of pseudo codes: complex, rigidly structured ones (action diagrams),from which you can generate code, and more informal ones that concentrate on a desciption of the solution of the problem.

More complex pseudocodes generally introduce a procedural dependency between processes, that is place them in a pattern where they can be executed (procedure action diagram).

The processes use the data model on item level:they refer explicitly to data items defined in the data model.

The processes are defined within a total process model where we can se how the processes relate to
- business procedures and dialogs
- other processes within the system
- other systems

The process model is the main input to program construction and coding.

Detailed business procedure analysis

All the business procedures identified earlier are built according to the principles also stated earlier.

We have, roughly speaking, two different ways of building business procedures depending on their potential for formalization.

Situations with a high potential formalization, generally more repetitive, "routine" kind of work flows like opening a new account, are built with a formal method including formal graphical documentation techniques.

Usually we form groups of people who build procedures in sessions lead by a session leader and supported by the graphic method. They build a sequence of activities and the information flowing in that sequence.

First we build the total procedure. Activities with computer support, either dialogs or activities where the computer produces reports etc., are isolated. When we have run through this a couple of times and the procedure as a whole is stable, we build the dialogs using another graphical method and prototyping/simulation.

Finally we specify all information passing between man and the computer on data item level on layouts.
We also relate this to the relevant automated processes needed, (se detailed function analysis).

Situations that are difficult to formalize as a sequence of activities are dealt with in a different manner. These sitations are normally not repetitive, but directly business decision oriented. The work done manually is often more complex and relies on more complex information, e. g. foreign exchange dealing, business analysis on trends, management decisions.

To come to grips with these situations we rely on the data model and dialog prototypes related to defined business situations e.g. dealing in a certain product.
The user often finds it natural to specify these situation via the information used. He often knows what information he uses when he makes the decision to "buy 20 million dollars" although he can't specify the processes leading to that decision.

So, given the data model, we systematically work our way through all business situations and prototype their dialogs.

We specify the dialogs on data item level and relate them to the process model.

The results from the detailed business procedure analysis are used
- as an input to the design of layouts
- to create instructions used for the operative business, both generally on how to run the business and specifically on how to use the computer support.

CLASSIFICATION OF CASE-TOOLS

The discussion that follows is based on the three-column matrix: "three dimensions of system development". The matrix is a simplified graphic version of what has been described earlier in this paper.

If we compare our three-column matrix with the functions found in different CASE-tools, how much does each tool cover?

First it should be noted that with very few exceptions the functions of different commercial CASE products are easily mapped onto the pattern. One such exception is the JSD Speedbuilder, whose entity life-cycle structures cover parts of all three columns, but leaves such important aspects as overview functional structure, entity structure and business procedures without support.

Data, function (no process logic) and screen-painting

There is a number of different CASE-tools that all cover
* the "data" column (sometimes including SQL code generation for first-cut databases),
* parts of the "function" column plus
* the screen-painting "slice" of the "event/procedure" column.

They all maintain links between different descriptions to varying degrees and are therefore often called "integrated".
Well-known representatives of this group are DEFT, DesignAid and Excelerator. The British AutoMate Plus product (see matrix) also falls into this category, but contains an embryotic dialogue flow function as well.
These products all use Yourdon/Gane&Sarson-like data flow diagrams for function modelling and some kind of Entity-Relationship modelling on the data structure side.

Data, function (process logic included) and screen-painting

The two well-known products that have taken "Information Engineering", as defined by James Martin, as their point of departure - the Information Engineering Facility (IEF) and the Information Engineering Workbench (IEW), see matrix - both cover the above defined area with an important addition: process logic definition in the form of pseudo-code with formal references to data usage. This means that they have built a foundation in their Business Analysis oriented modules for later code generation - and in fact they both offer procedure code generators (mandatory for the IEF customer, optional for the IEW customer) that use this process logic as a base to which technical information is added.

Data only

There are some products that are built to exclusively support development within the "data" dimension. Here we find some of the forerunners of CASE technology - products like MSP:s Design Manager and Data Designer, an elderly relative to the IEW. The Janus product is a mainframe-based member of this group from the time before the "PC graphics revolution", just like the other two mentioned above. The Bachman Data Analyst tool and the Scandinavian tool Modellator are more recent, PC-based products for the "data" dimension, taking advantage of PC graphics. At least the Bachman product aims at generating database description code.

Function only

The Swedish ISAC methodology has through the years been supported by different tools, many of them spread in rather narrow circles. The most well-known is GraphDoc, supporting the structural aspect of the "function" dimension.

Event/procedure only

The "event/procedure" dimension is the most poorly supported by the leading "integrated" CASE-tools. Apart from screen design there is almost no support (Excelerator's "presentation graphs" can be used to graphically describe business procedures. But since Excelerator doesn't know what business procedures are there is no real CASE-support for them).
A few organizations, like the ABN Bank in the Netherlands, have built their own free-standing software to support this dimension. However the only commercially available product we know of that supports business procedure description (in combination with dialogue flows) is the Swedish product RUTH, which is specifically built for this purpose. The business procedure concept seems to be part of a Scandinavian/Dutch tradition, which might explain why it is not supported by any of the major "integrated CASE" suppliers, which all are Anglo-Saxon.

How to choose

When you want to select a CASE-tool and start talking to the different CASE vendors you find that each vendor has its own language, incompatible with the others. You need a selection method that is independent of the vendors, a "measurement system" that is neutral.

The matrix forms a good basis for understanding what different CASE-tools do and what they don't. This should be considered first, in our opinion, before you evaluate how neatly and effectively they do what they do.

Based on the matrix it is easy to see that no commercially available CASE-tool covers all the functions needed. There are a few products that cover most of the matrix, but if you want to develop information systems the way we recommend you have to either buy an auxiliary product or build certain parts yourself.
This is true even if you just consider systems analysis work.

In fact that is what has happened in most organizations where we have been involved. One of the integrated "function/data-oriented" products has been purchased as a foundation, on which one or more extra products have been added. In these cases the matrix has served as a base for the evaluation work.

INTRODUCTION OF CASE-TOOLS

The CASE technology in its present form doesn't change the basic ideas on how systems analysis and design should be conducted.
A CASE-tool is only a piece of software that helps you do what you should have done anyway. The underlying techniques have all been known for years.

But CASE technology
* makes it much easier to administer all the different descriptions. We can now work the way we should have done before - without becoming overloaded with paperwork. It is even easier in many cases to follow the rules than to break them.
* makes it possible to maintain links between different objects in our models. It thus becomes possible to raise the level of ambition towards building the "complete logical system" that we mean should be the result of system analysis.

Often CASE technology is introduced in an organization one project at a time. We have found that by building the structural models we mentioned earlier (function model, entity-relationship model and "trend-setting" business procedure description) one at a time and entering each model in the CASE-tool just after it has been created
* we only have to teach the project members the part of the tool that is relevant to that model
* we only have to teach them the tool itself, not what to do with it
* the CASE-tool is regarded by its users as "just" a normal computer system to support work they thoroughly understand (after the seminars, that is), and not as a piece of magic

What if you want to introduce CASE technology in an organization on a large scale? We believe that in order to take advantage of CASE technology an organization has to reach a certain level of "maturity". To do this it is normally a good idea to
* first ensure that there is a common understanding within the organization about how systems analysis and design should be performed
* then introduce the CASE-tools on project level, and thereby establish how the CASE-tool use should be tailored according to the characteristics and traditions of the organization and
* last move on to the more difficult tasks of coordinating structural information as well as reusable "pieces" (processes, entities and so on) on a company-wide basis.

Three dimensions of system development

	FUNCTIONS	*DATA*	*EVENTS/ PROCEDURES*
ANALYSIS, STRUCTURES	**Function model**	**E–R model**	**Principles for business procedures**
ANALYSIS, DETAILS	**Detailed processes**	**Data model (normalized)**	**Business procedures Dialogues**
TECHNICAL DESIGN	**Designed programs**	**Designed databases**	**Designed dialogues and physical layouts**
CONSTRUC–TION	**Coded programs**	**Coded databases**	**Business procedure instructions**

Hernbäck & Lindqvist

Three dimensions of system development – *AutoMate Plus*

	FUNCTIONS	*DATA*	*EVENTS/ PROCEDURES*
ANALYSIS, STRUCTURES	Function model	E – R model	Principles for business procedures
ANALYSIS, DETAILS	Detailed processes	Data model (normalized)	Business procedures Dialogues
TECHNICAL DESIGN	Designed programs	Designed databases	Designed dialogues and physical layouts
CONSTRUCTION	Coded programs	Coded databases	Business procedure instructions

Three dimensions of system development – IEW and IEF

	FUNCTIONS	*DATA*	*EVENTS/ PROCEDURES*
ANALYSIS, STRUCTURES	Function model	E – R model	Principles for business procedures
ANALYSIS, DETAILS	Detailed processes	Data model (normalized)	Business procedures Dialogues
TECHNICAL DESIGN	Designed programs	Designed databases	Designed dialogues and physical layouts
CONSTRUCTION	Coded programs	Coded databases	Business procedure instructions