GoldminePower: Decisions in Governmental Process Design

Evert van Heel* Tom M. van Engers†
*postbus@e-vanheel.demon.nl
†Leibniz Center for Law,
Faculty of Law, Universiteit van Amsterdam
Amsterdam, The Netherlands
vanengers@lri.jur.uva.nl

Abstract. The success of governmental policy depends much on the success of governmental services. Governmental services in its turn depend much on governmental processes. Governmental processes therefore are a determining factor for the success and especially the justification of governmental behaviour. Process design is an important means of realising good en well-performing governmental processes. Governments raise specific demands and wishes in the field of process design, such as transparency and justification of policy. The process design method applied should consequently reflect these important aspects of governmental process design.

A reasonable assumption is that quality of (governmental) processes is related to the degree in which the involved process design decisions have been made explicit, and the degree in which the explicit decisions have been accounted for. In this paper we introduce a proposition for a new governmental process design approach called GoldminePower. This approach has the specific governmental requirements (the process's legal bases and the need for transparency and justification) as starting point. A strong focus will be put on explicit design decisions during governmental process design. The application of this method is not limited to public administration. Non-governmental organisations could also benefit from more transparent and justifiable process design.

1 Governmental Issues[1]

In this section we would first like to stress the importance of processes for governments. We do this by showing almost all governmental problems are process-related. The importance of processes for governments implies the importance of process design. That is why in the second sub-section we consider demands and wishes governments could and should have in the field of process design.

1.1 Process Related Governmental Problems

Most well-known problems with governmental policy tend to be process related. This becomes clear when studying a recent Dutch manifest of the committee "Belgen doen het beter [1]" ("Belgians perform better") in which the success of Dutch governmental policy is compared to the policy of other European states. In table 1 an overview of Dutch governmental problems from this manifest is given.

[1] ≪Johnny's in the basement / Mixing up the medicine / I'm on the pavement / Thinking about the government . . . ≫ (Bob Dylan – Subterranean homesick blues)

Evert van Heel and Tom M. van Engers, 'GoldminePower: Decisions in Governmental Process Design' in D. Bourcier (ed.), *Legal Knowledge and Information Systems. Jurix 2003: The Sixteenth Annual Conference*. Amsterdam: IOS Press, 2003, pp. 71-80.

Table 1: Some Dutch governmental problems

NR	Problem Kind	Problem	Description / Background	Possible process aspects
1	Budgeting	Huge costs of Dutch social security	The social security institute "Uitvoeringsorgaan WerknemersVerzekeringen"(UWV) is very expensive. This institute is the consequence of a merger between a large number of similar institutes for unemployment and illness insurance. The services of the UWV are carried out in processes that stem from those former institutes and that mainly have been kept unchanged.	Not integrating and (poor) maintaining redundant processes looks to be the cause. BPR necessary
2	Planning	Waiting lists in health care	The waiting lists in the Dutch health care institutes are notorious. There are many supposed causes amongst which lack of data exchange between institutes.	Unclear and inconsistent policy and unclear responsibilities of the different parties that are involved in the underlying services / processes might be a cause. BPR necessary
3	Quality	Citizens safety	The solve percentage of crimes in Holland is low. This is stated to be the main cause why a huge percentage of the Dutch citizens don't feel safe. One of the causes is said to be the lack of data exchange between police departments.	Crime resolving processes need to be improved BPR necessary
4	Quality	Errors in health care	Research has shown that on a yearly base about 90.000 hospital intakes are due to avoidable medication errors. These errors could have been avoided if the doctors had access to all of the patients' medical data	Processes need to be improved BPR necessary
5	Quality	Food safety	Holland lately has suffered a lot of food crises like the BSE-crisis the Pigs-plague, the Foot-And-Mouth-crisis and the Birds-plague. There is a need for more transparency in the food chain processes here.	Different services / processes have different goals and don't work together very well BPR necessary
6	Quality	Dealing with terror	Data exchange between for instance the Dutch intelligence service and the police departments is no good.	Terror preventing and resolving processes need to be improved BPR necessary

This list is far from complete and could easily be extended with other Dutch and non-Dutch governmental problems. In most cases it will be easy to link those other governmental problems to process shortcomings as well. So maybe the best way for governments to improve their governing is improving their governmental processes.

1.2 Governmental Demands and Wishes in the Field of Process Design

In the previous sub-section we showed governments have a lot to gain by good process design. This can lead to well-performing processes and solved governmental problems.

Now we will consider the demands and wishes governments should have in the field of process design. In section 2 we'll unfold a design method that satisfies these demands and wishes. As governmental demands we consider requirements that stem from legislation or that are for other reasons specific for governments, wishes stem from other sources like good management. In table 2 our list is displayed.

2 GoldminePower Method

The central idea in this paper is to address the issues from section 1 with a method that combines two promising methods from literature. Both methods "Goldmine"[2] and "Power"[3] make the same kind of claims: "Better, cheaper and faster" than common practice. We observe that current process modelling approaches are focussed on business process redesign issues and optimization of existing processes rather than establishing a transparent representation of the temporal aspects that are essential to law enforcement and other forms of legal

Table 2: List of governmental requirements for process design

Requirement	Description
Legal base and justification	For all governmental actions (this includes every step in a governmental process) there must be a legal base, so that they can be justified.
Transparency	There are no specific juridical reasons to strive after transparency (governments must be able to justify a specific case, but then it is sufficient to refer to the involved legislation). But although governments in general don't have to justify process design decisions and the working of its processes, good reasons exist for governments to indeed strive after transparency. The first reason is that the governmental processes can be justified to politics, which will create support for the government's policy. The second reason is that case-justification can be faster, cheaper and easier done, since any process step in a transparent process knows its origins in legislation
Effectiveness	"Obtaining as much as possible effect with as little as possible effort" It is good management to strive after effectiveness. Just like businesses governments have limited budgets. In process design achieving more effectiveness means that indicators must play an important role. It must be clear which indicators represent the effects we want to measure, how these indicators must be measured, which results are expected, etc.
Efficiency	"Obtaining all required goals at the right time" This is also important in good management. If not enough budget is available this may conflict with the effectiveness wish. In process design terms efficiency has a lot to do with the sequence order in which process steps have to be carried out, so it is important that design decisions about this are made explicit.
Flexibility	Legislation in democratic countries is very often subject to change. It is therefore very important to governments that they can easily change their services and processes that are results of legislation. In process design to achieve flexibility it is very important that all design decisions that have led to a process implementation have been stored so that they can be re-used.

reasoning.

Besides the other things we are convinced to be lacking in current process design approaches this feature, being able to contain transparent relationships between process models and the legal sources it originates from, is essential for a modelling approach applicable to the governmental processes.

First we have to say something about the scope of the new combined "GoldminePower" method as compared to the Power method. The Power method focuses on declarative aspects in source text (For instance: "A person older than 17 is an adult".).

The Power method does not focus on procedural and temporal aspects in the used source texts. The final integrated Power model in UML/OCL format therefore needs extra procedural knowledge when it is used to create an application. Such an application for instance needs to know what its goals should be; this information should be available in the design of the business process in which the application that is to be developed plays a role. The GoldminePower method focuses on procedural aspects to use them for process designs. So GoldminePower can perfectly be used additional to the Power method.

In this section the combined method GoldminePower is explained in a step-by-step manner.

2.1 From Legislation to Legislation Design Decisions

The way legislation documents are transformed to a formal model in GoldminePower is very similar to way this is handled in Power. Consider the example in figure 1.

The first step in the Power method is to register the structure of the documents. In GoldminePower structure blocks and references in legislation documents are found in exactly the same way as is done in the Power method. In the example the structure block kinds "Law", "Chapter", "Article" and "Sentence" are found.

After this the text in the sentence structure blocks is scanned for process design decisions which are registered in the Goldmine business knowledge format. More specific the text is scanned for:

Law Y
Chapter 1
Article 1.1 Intentions and definitions
This law is meant to make people happier.
A person older than 17 is an adult.
Article 1.2 X
Anyone can order the town hall to perform X for him or her.
The town hall floor manager then will do A.
B will also have to be carried out.
B may not be done before A is finished.
...

Figure 1: A legislation example

- Existence of process (sub) steps
- Properties of process steps (data, triggers, indicators, actors, . . .)
- Order relations of steps
- Priorities of indicators on process steps (policy rules)

This scan is additional to the scan that is used in the Power method. Where the Power method is constructing an ontology using an OCL/UML notation and therefore is noun-oriented (Types are mostly nouns), the GoldminePower is verb-oriented (Steps are mostly verb-phrases).

The thus created “Translated” GoldminePower design decision models are Power-like combined into one “Integrated” GoldminePower legislation design decision model. Just like in the Power method these steps can be supported by natural language processing software [4]. Also just like in the Power[3] method found unclearness in legislation can serve as feed-back to legislation drafters.

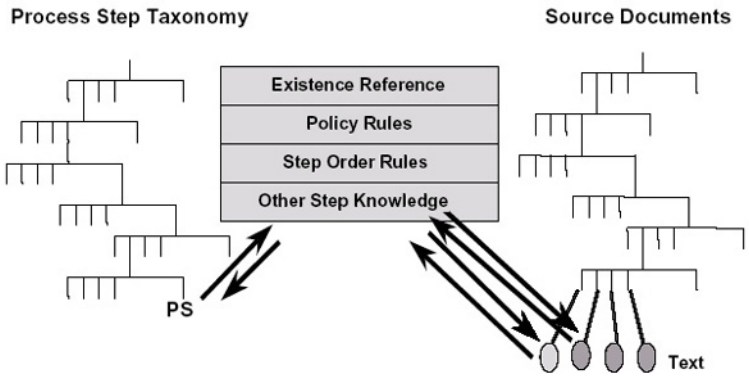

Figure 2: View on the Goldmine business knowledge representation

In figure 2 a view on the Goldmine “Business knowledge” is given. The Goldmine claim is that all that is needed to implement a (workflow) process can be stored in this business knowledge representation and that this knowledge is easy to maintain, especially if also traceability to and from source documents is realised.

This format is in the GoldminePower method used for the “design decision models”. The decisions are located in a taxonomy of process steps. In table 3 we can see a design decisions model for process X from figure 1.

As the reader can see the design decisions are easy to understand and to maintain.

Table 3: A design decisions model

Design Decision Nr	Decision Kind	Content	Parent	Source /Why
1	Step existence	X is a governmental process		Law Y, Article 1.2, first sentence
2	Step existence	A is a sub-step	X	Law Y, Article 1.2, second sentence
3	Step existence	B is a sub-step	X	Law Y, Article 1.2, third sentence
4	Order of sub-steps	B must be carried out unconditional after A	X	Law Y, Article 1.2, fourth sentence
5	Other step knowledge	Public_Order is a trigger	X	Law Y, Article 1.2, first sentence
6	Other step knowledge	Happiness is an indicator of X	X	Law Y, Article 1.1, first sentence
7	Other step knowledge	Happiness-after must be greater than happiness-before	X	Law Y, Article 1.1, first sentence
8	Other step knowledge	The town hall floor manager is the actor	A	Law Y, Article 1.2, second sentence
....				

2.2 Supporting Legislation Drafters Using a Generated Process Design from Legislation Design Decisions

Each single design decision from the legislation design decisions model is easy to understand for any reader. However, to see what the effect of all business knowledge as a whole is this model is no good. A Petri-net [5] like semantic network is much better equipped to serve as a validating view. This is also known from a study on formal representation format [6] where scenarios prove to be better equipped to validate a model than rules. And this is probably why drawing/case tools (ABC, Visio, Rose, Mega ...) process simulators (Testbed, bWise,...), and workflow engines (Staffware,..)use formal network representations for process design models [7].

In the Goldmine method is the knowledge that is needed for constructing a formal network representation from out of the design decisions is called ”modelling knowledge”. This typically is the knowledge that a human process designer uses to apply new facts from documents or experts in the process design. In doing so there is a risk that the facts that have led to the process design are not registered. A slight change in those facts might result in a big change in the formal network representation(s). When design decisions are not present in an explicit format Business Process Redesign can be a big effort.

The human process designer’s task must not be under-estimated. But since modelling knowledge is almost constant (It only changes if important modelling conventions change) it is something that can be done by a computer. This is one of the main ideas in the Goldmine project. In Goldmine an algorithm is described for producing a process design in an all-information-containing network representation (The “integrated process description” in figure 1) from out of a design decisions model. A prototype that applies such an algorithm was not included in the Goldmine research but was used nevertheless. This prototype also can export views on the design for MS Visio[7].

The existence of a very good version of such a design generator is essential to the Goldmine and GoldminePower methods. In the rest of section 2 the existence of a perfect generator is assumed. Starting from a design decisions model this generator can produce (any model view on) a process design in less than 1 second.

In figure 3 a view on the meta model for a process design environment is given.

The dark blue (dark grey) rectangles in figure 3 represent abstract classes. The light blue (medium grey) rectangles represent knowledge objects that are present in the design decisions

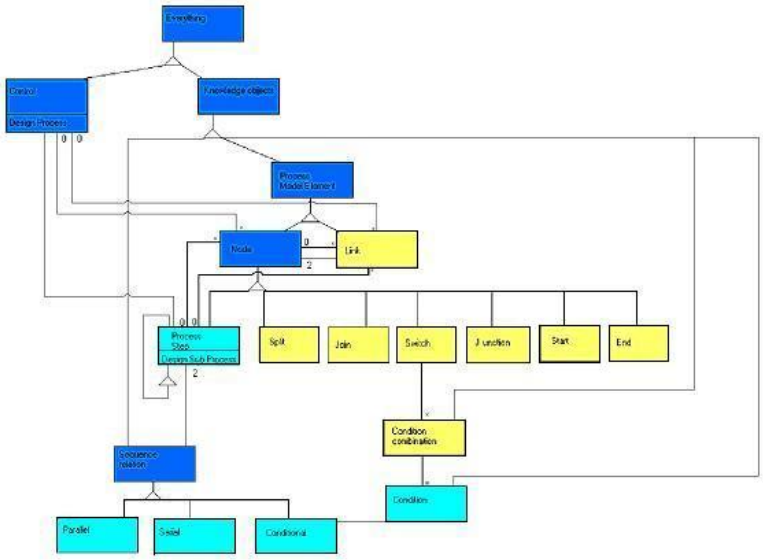

Figure 3: A view on the meta model for a process design environment

model. The yellow (light grey) rectangles represent objects that are generated because they are needed for a formal network representation.

Having different views on a process design is also a very important aspect of Goldmine and GoldminePower. The town hall floor manager would like to see a view on process X that gives his own work in a very detailed format, and all other steps in a global format. He has got an extra requirement saying he would like to see all town hall floor activities “swim-laned”. A workflow engine will want another kind of view as an interface, and is not interested in details that are not workflow-supported.

Views can of course also differ in graphical knowledge representation (See figure 4)

Here you see three examples of style views in which formal network descriptions can be displayed and which the perfect generator is able to deliver. The left-hand style view represents the format for the integrated process descriptions that is used in the Goldmine prototype.Characteristics of the left hand style representation:

- A process step always has one incoming and one outgoing flow
- Decomposition like in IDEF3 [8]
- Unlike IDEF3, but after Brand & v.d. Kolk [9], XOR (switch/junction) and AND (split/join) objects. IDEF3 uses OR and AND. The OR has the drawback that the number of flow instances is uncertain.
- Level start and end markers (Can be suppressed in the Goldmine prototype)
- Dummy process steps (Can be suppressed in the Goldmine prototype)

The top style view represents the CSC Catalyst [10] process view representation. Finally the right-hand view represents a style one could adopt after studying Brand & v.d. Kolk.[9]

Now back to supporting legislation drafters. From the legislation design decisions model a legislation process design can be generated. When performed by a perfect generator this action only fails if inconsistent design decisions exist.

An example is given in Table 4. These inconsistent design decisions are of course important input to legislation drafters. In most cases however, the generator will be able to make an

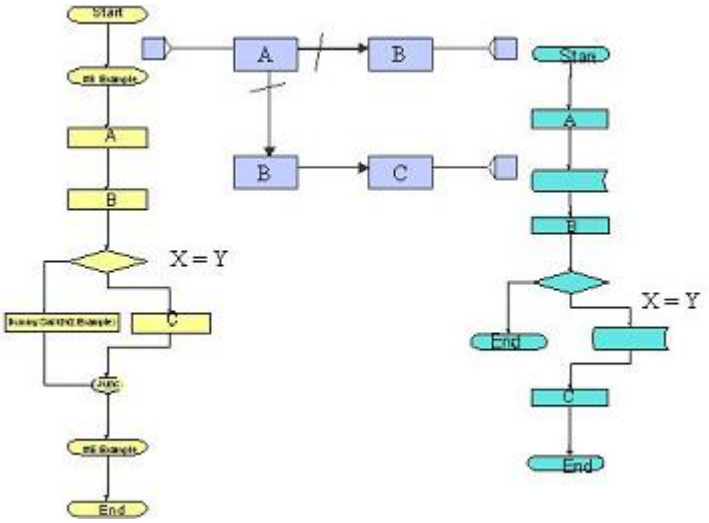

Figure 4: Possible graphical view representations for the same design decisions model

Table 4: An example of inconsistent design decisions

Design Decision Nr	Decision Kind	Content	Parent	Source /Why
1	Step existence	X is a governmental process		Law Y, Article 1.2, first sentence
2	Step existence	A is a sub-step	X	Law Y, Article 1.2, second sentence
3	Step existence	B is a sub-step	X	Law Y, Article 1.2, third sentence
4	Order of sub-steps	B must be carried out unconditional after A	X	Law Y, Article 1.2, fourth sentence
5	Order of sub-steps	A must be carried out unconditional after B	X	Law Z, Article 3.2.1

integrated process description for which relevant views can be exported to and validated by a case tool or a process simulation tool (like Testbed, bWise . . .)[7].

Note that the generator includes all modelling knowledge. If for instance in table 4 decisions 4 and 5 are not present the generator must implement a default order for sub-steps A and B. (Simultaneously, In alphabetical order, . . .).The most important feedback for legislation drafters we expect will come from optically reviewing the generated process models. A human legal expert can for instance see that process steps are missing, inconsistent, redundant or in an illogic order. The involved legal expert will get his views in a format to which he is most familiar [11]. His remarks will lead to a changed set of design decisions and afterwards in changed design model views.

For further validation a simulation tool can produce valuable information for legislation drafters. If for instance change in legislation was meant to generate a cheaper governmental service, a process design view in simulator environment might show the changed process is more expensive instead of cheaper.

2.3 Adding Local Design Decisions

(Parts of) government processes might be carried out in a local institute that has its own policy, wishes and demands with regard to its processes. To get the total process design this business knowledge has to be added to the legislation design decisions model. This can again be done by scanning documents with institutional policy, but also by making new design decisions

based on business economical considerations or common-sense. In table 5 we skipped the incorrect design decision 5 from table 4 and added two local design decisions. This seemed to make sense after consulting some town hall documents and a local town hall floor manager.

Table 5: Added local business knowledge

Design Decision Nr	Decision Kind	Content	Parent	Source /Why
1	Step existence	X is a governmental process		Law Y, Article 1.2, first sentence
2	Step existence	A is a sub-step	X	Law Y, Article 1.2, second sentence
3	Step existence	B is a sub-step	X	Law Y, Article 1.2, third sentence
4	Order of sub-steps	B must be carried out unconditional after A	X	Law Y, Article 1.2, fourth sentence
5	Step existence	"Sending a welcomes gift" is a sub-step	X	Institutional policy, page 1, third paragraph
6	Order of sub-steps	"Sending a welcomes gift" is carried out Parallel with A	X	Because A is the first official step and we don't want any delay in the total process
7	…..	…	..	..
…				

After all local business knowledge has been added the total design decisions model is finished.

2.4 Validating the Total Set of Design Decisions Using a Generated Process Model

In figure 5 a possible view is given on a generated process model from the design decisions model in table 5. This kind of graphical representation is the kind that is used in the Goldmine thesis (see 2.2.). Within the same graphical representation kind still a lot of views on a process model design are possible depending on the perspective the reader (Man or machine) is interested in. Possibly step A has been decomposed whereas the reader is not interested in the details of step A.

Just like in 2.2. experts have to look at the total design model views to detect mistakes or less optimal design decisions (Maybe an unconditional sub-step of A which existence is based on a legislation design decision has the same purpose as a conditional sub-step N (Based on a local design decision) of X that occurs after B.

A simulator can help validating the behavior of a process model. If from legislation design decisions are present saying TotalTime is an indicator for this process and the maximum TotalTime is 4 days, and from local design decisions it is clear that step A on an average will cost 1 day and step "Sending a welcomes gift" on an average will cost 2 weeks, the simulator might show that the process model does not satisfy the governmental demands.

In this case the total design decisions model could be improved by skipping step "Sending a welcomes gift" as a sub-step of X, and adding a design decision on step A saying step A is a trigger for a local process called "Sending a welcomes gift".

2.5 Implementing the Validated Process Model

After the process model has been designed and validated in an iterative way the generator can provide a workflow engine tool[7] interface with a view on the process model that is necessary that makes sense to the workflow engine. During execution of any process step of

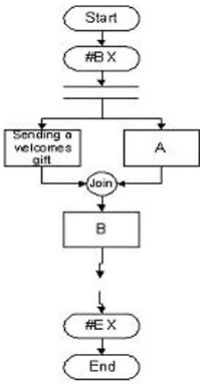

Figure 5: Possible design view

the workflow process the source of the design decisions that have led to or influenced the have current task can be re-found and re-used.

For the more difficult tasks the process can be supported by modules that have been developed with the Power[3] method. The implemented process model has to be monitored on its indicators so that for instance the intended mean values can be re-evaluated. This might lead to a changed design decisions model, and afterwards to a better process model design and a better performing implementation.

3 Research Plans

We plan to test two major hypotheses which are surrounded by many questions and definitions. The first hypothesis we feel needs testing is:

H1: *The degree of (accounted for) explicit design decisions is an indicator for process quality.*

We hereby give the most important definition: Explicit design decision == *a decision about the existence of a process (sub step), about the order in which sub steps of a process (sub step) have to be carried out, or about a property of a process (sub step) that is explicitly mentioned in the process design*. For the H1 research we will investigate as many as possible governmental processes. Possible design decisions formats will be part of the research. Research after process quality indicators will also be done. For this the list of judgement criteria[2] for process designing that is a result from the Goldmine project will be a starting point.

The second and last hypothesis that will be tested is:

H2: *The GoldminePower method for implementing (changes in) governmental policy into governmental processes should be widely recommended.*

We would like to try the GoldminePower method on a real project that is initiated by a planned change in legislation. This has the advantage that the quality of the involved governmental process can be measured before and after the business process redesign project. The research for H2 will include all possible computer support to the method.

4 Conclusions

Governmental problems are mostly process related, and might be solved when the involved processes are improved by means of excellent process design that satisfies all governmental requirements in the field of process design. The GoldminePower method promises to provide this. Governments could get more grip on their process related problems because of the transparency brought by the method. All decisions that lead to a process implementation are made explicit and are accounted for. This makes processes and changes in legislation easy to maintain. Design decisions can also include decisions on indicators for which the process has to be optimised thus helping to create optimal performing processes. Process designs could be validated even by politicians since the GoldminePower method can provide audience centred views[11] on process designs.

The GoldminePower method therefore is a very promising set of ideas. Because of the great interest governments have in solving their problems it is very important that the proposed method is researched as soon as possible. We hope governments will be very eager for supporting the planned GoldminePower research.

References

[1] February 2003,Een kwestie van uitvoering, Vernieuwingsagenda voor de presterende overheid, site http://www.kennisland.nl/kennisland/binaries/KLsite/actueel/publicaties/Een_kwestie_van_uitvoering.pdf

[2] van Heel,E.J.,2000,Goldmine-A KT approach to workflow process designing, MSc thesis Cibit, Utrecht

[3] van Engers,T.M., Kordelaar,P.J.M., den Hartog, J., Glassée, E.,2000, POWER: Programme for an Ontology based Working Environment for modeling and use of Regulations and legislation. In proceeding of the 11^{th} workshop on Database and Expert Systems Applications (IEEE) Eds. Tjoa, Wagner and Al-Zobaidie, Greenwich London, ISBN 0-7695-0680-1, pp. 327-334.

[4] Ron van Gog, Tom M. van Engers,2001,Modeling legislation using natual language processing, Proceedings of the 2001 IEEE Systems, Man, and Cybernetics Conference, available at http://www.lri.jur.uva.nl/∼epower/publications/modleg.pdf

[5] van der Aalst,W.M.P.,1996,Petri-net-based Workflow Management Software, Proceedings of the NFS Workshop on Workflow and Process Automation in Information Systems, Athens, Georgia

[6] Tom van Engers, Liesbeth van Driel and Margherita Boekenoogen,2002, The effect of formal representation formats on the quality of legal decision-making,Legal knowledge and information systems-JURIX 2002: The fifteen annual conference,ISBN 1-58603-299-2, page 63

[7] W.D. van Dommelen, S.M.M. Joosten, M.C.J. de Mol, H. de Zwart,1999,Vergelijkend onderzoek hulpmiddelen beheersing bedrijfsprocessen, EAP, Apeldoorn, downloadable from: http://www.anaxagoras.com/publicaties/vohbb.htm

[8] Richard J. Mayer,.Christopher P. Menzel, Michael K. Painter, Paula S. deWitte, Thomas Blinn, Benjamin Perakath, 1995, IDEF3 Process Flow and Object State Description Capture Method, full report downloadable from: http://www.idef.com/idef3.html

[9] Brand, N.A. & van der Kolk, J.R.P., 1997, Werkstroomanalyse en –ontwerp, Het logistiek vriendelijk ontwerpen van informatiesystemen, Kluwer bedrijfsinformatie, ISBN 90 267 2100 5.

[10] CSC process thread, see Allen, P., & Frost, A. ,1998, Component-Based Development for Enterprise Systems: Applying The Select Perspective, Cambridge University Press, ISBN 0-521-64999-4 paperback.

[11] Abeysinghe, G.K., Henderson, P., Phalp, K.T. and Walters,>=1997, R. An Audience Centred Approach for Business Process Reengineering, see http://www.ecs.soton.ac.uk/∼rjw1/papers/method/method.pdf