Case Handling in Construction

W.M.P. van der Aalst
Department of Information and Technology, Eindhoven University of Technology, PO Box 513, NL-5600 MB, Eindhoven, The Netherlands. w.m.p.v.d.aalst@tm.tue.nl

M. Stoffele
Architectural Design Management Systems (SAI), Eindhoven University of Technology, PO Box 513, NL-5600 MB, Eindhoven, The Netherlands.
Heijmans IBC Bouw PO Box 6249, NL-5600 HE, Eindhoven, The Netherlands.
mstoffele@heijmans.nl

J.W.F. Wamelink
Department of Information and Technology, Eindhoven University of Technology, PO Box 513, NL-5600 MB Eindhoven, The Netherlands. j.w.f.wamelink@tm.tue.nl
Infocus Management Consultants b.v., PO Box 555, NL-4100 AN, Culemborg, The Netherlands.

Abstract
Case handling is a new means for supporting flexible and knowledge intensive business processes. Unlike workflow management, which uses predefined process control structures to determine what *should* be done during a workflow process, case handling focuses on what *can* be done to achieve a business goal. In this paper, case handling is introduced as a new possibility for supporting construction processes. The construction of buildings and related facilities is a difficult and complex process, which requires both support and flexibility. This paper describes the application of the case-handling principles within Heijmans. Heijmans is one of the leading companies in the Dutch building industry and interested in IT support for their construction processes. We have used the case-handling system FLOWer to provide automated support for preparing the construction of complex installations. In this paper, we report our experiences.

Keywords: Case handling, process control, process modeling, workflow management, process improvement strategies

1. Introduction

Construction processes are notorious for their complexity and changes during the process [10,19,25]. Many attempts to provide automated support for these processes have failed. Today's processes in manufacturing, logistics, and the service industry are supported by information systems. These systems help workers in monitoring, executing, and controlling business processes. This support is enabled by rigorously structuring the processes. Automated support of business processes typically improves performance (e.g., reduced flow times and increased throughput), reduces labor costs, and increases quality (e.g., less errors). Given these observations, it remains a challenge to apply these systems and principles to construction processes in the building industry.

Nowadays, many administrative processes are supported by workflow management systems. Workflow management systems such as Staffware, IBM MQSeries Workflow, COSA, etc. offer generic modeling and enactment capabilities for structured business processes. By making graphical process definitions, i.e., models describing the life cycle of a typical case (workflow instance) in isolation, one can configure these systems to support business processes. Besides pure workflow management systems many other software systems have adopted workflow technology. Consider for example ERP (Enterprise Resource Planning) systems such as SAP, PeopleSoft, Baan, Oracle, as well as CRM (Customer Relationship Management) software. Despite its promise, many problems are encountered when applying workflow technology. As indicated by many authors, workflow management systems are too restrictive and have problems dealing with change [4,6,7,11,13,15,17,18,27]. Many workshops and special issues of journals have been devoted to techniques to make workflow management more flexible [4,6,17,18]. Some authors stress the fact that models should be as simple as possible to allow for maximum flexibility [7]. Other authors propose advanced techniques to support workflow evolution and the migration of cases of one workflow model to another [11,27]. If the process model is kept simple, only a more or less idealized version of the preferred process is supported. As a result, the real run-time process is often much more variable than the process specified at design-time. The only way to handle changes is to go behind the system's back. If users are forced to bypass the workflow system quite frequently, the system is more of a liability than an asset. If the process model attempts to capture all possible exceptions [24], the resulting model becomes too complex to manage and maintain. These and many other problems show that it is difficult to offer flexibility without losing control.

In this paper, we focus on the application of workflow technology to construction processes. Given the fact that contemporary workflow management systems such as Staffware and IBM MQSeries Workflow have problems providing operational flexibility, it does not make sense to try and apply these systems to construction processes. Therefore, we propose an approach based on the case-handling paradigm [3]. This paradigm is supported by a case-handling system named FLOWer [22]. We consider construction processes in the building industry the acid test for case-handling.

The work reported in this paper, is the result of a project conducted within Heijmans Bouw. Heijmans Bouw is part of Heijmans N.V. that operates in the construction industry and related industries. The main activity of Heijmans Bouw is the realisation of buildings. Heijmans Bouw represents Heijmans N.V. in all sectors of the private and business housing. Figure 1 is used to describe the scope of the project.

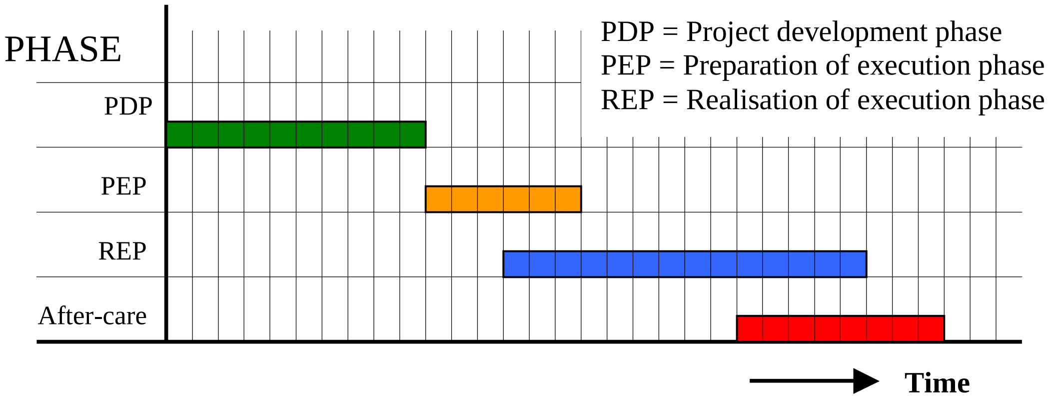

Figure 1: phases in projects

Heijmans Bouw divides its projects into four phases (Figure 1): Project Development Phase (PDP), Preparation of Execution Phase (PEP), Realization of Execution Phase (REP), and after care. The main focus of Heijmans Bouw is on the execution, i.e., PEP and REP. Especially in the PEP management and control of the project is important and can be very effective for the whole project. For managing its projects Heijmans Bouw uses so-called project manuals. An example is the PEP manual. This manual contains standard documents and schedules and has been an important starting point for the work presented in this paper. It should be noted that in spite of the existence of project manuals, no computer support for controlling and managing any of the four phases shown in Figure 1 existed.

The research project described in this paper focuses on the application of the case-handling approach in the PEP. We have applied this approach to the preparation process of semi-prefab concrete floor elements and the preparation process of HVAC (Heating, Ventilation and Air-conditioning) installations. Automation seems particularly useful for these two processes. To find out the possibilities of automation for the selected processes and to get inside the processes, a process model has been developed. Important aspects in making the process model are the required flexibility and the level of detail in defining the activities. Both aspects are needed to be able to customize the process model to multiple projects. If the activities are defined in too much detail, the process model becomes difficult to adjust and re-use is limited. However, by defining the activities at a too high level, the process model cannot be used as a stepping-stone for a system that really supports the PEP.

The process models of the preparation process of semi-prefab concrete floor elements and the preparation process of HVAC installations indicated that these processes could be automated. The required flexibility is translated by modeling loops and optional routings in the process model. The level of detail in defining the activities is comparable to that used in the project manuals. Given prior experience with the manual, this is acceptable. Based on these experiences a prototype was developed to clarify the possibilities of a case-handling system for the two processes. The prototype enabled a discussion about the possibilities of such a system. Some workers of Heijmans Bouw were asked to test and to evaluate the prototype. The results of the test were evaluated and translated to the complete PEP. The reactions on the prototype were mainly positive.

The remainder of the paper is organized as follows. First we introduce the case-handling method. Then we discuss the suitability of this method in the building industry. Based on this we describe the case study conducted within Heijmans. First, we discuss the process model for the considered process parts. Then, we describe the prototype and the experiences with the prototype. Finally, we conclude with general observations.

2. Case handling

Before we continue, we want first to introduce some standard workflow terminology. Workflow management systems are *case-driven*. This means that only business processes describing the handling of one *case* (workflow instance) in isolation are supported. Many cases can be handled in parallel. However, from the viewpoint of the workflow management system these cases are logically independent. To handle each case, the workflow management system uses the corresponding *workflow process definition*. The process definition describes the routing of the case by specifying the ordering of *activities*. Activities are the logical units of work and correspond to atomic pieces of work, i.e., each activity is executed by one worker (or another type of resource) and the result is either “commit work” or “abort and roll back”. Typically to specify the ordering of activities graphical languages such as Petri nets [1] or workflow graphs [27] are used. These languages allow for sequential, conditional, and parallel routing of cases. Some of the workflow management systems allow for more advanced constructions [5]. Typically an activity, which is enabled for a given case, may be executed by many workers and many workers may execute a given activity. To support the distribution of work, the concept of a *role* is used. A worker can have multiple roles, but an activity has only one role. If activity A has role R, then only workers with role R are allowed to execute activities of type A. Based on this information, the workflow management system works as follows: The corresponding workflow process definition is instantiated for each new case, i.e., for each case (e.g., request for information, insurance claim, customs declaration, etc.) a new workflow instance is created. Based on the corresponding workflow process definition, the workflow engine calculates which activities are enabled for this case. For each enabled activity, one work-item is put in the in-tray of each worker having the appropriate role. Workers can pick work-items from their in-tray. By selecting a work-item the worker can start executing the corresponding activity, etc. Note that, although a work-item can appear in the in-tray of many workers, only one worker will execute the corresponding activity. When a work-item is selected, the workflow management system launches the corresponding application(s) and monitors the result of executing the corresponding activity. Note that the worker only sees work-items in his/her in-tray and when selecting a work-item only the information relevant for executing the corresponding activity is shown.

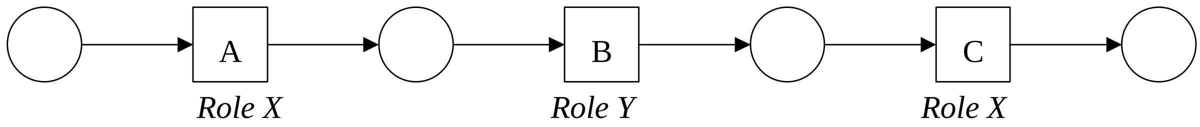

Figure 2: a simple process consisting of three activities

To illustrate the basic workflow concepts consider Figure 2. This figure shows a simple process consisting of three activities (A, B, and C) which are executed sequentially. First activity A is executed, then activity B is executed, and finally C is executed. Activities A and C are executed by a person with role X. Activity B is executed by a person with role Y. Note that the activities are connected by circles (often referred to as places) which correspond to the states in-between two subsequent activities.

In this paper, we argue that the lack of flexibility in contemporary workflow management systems stems from fact that routing is the only mechanism driving the case, i.e., work is moved from one in-tray to another based on pre-specified causal relations. This causes the following problems:

- Work needs to be *straight jacketed into activities*. Although activities are considered to be atomic by the workflow system, they are not atomic for the user. Clustering atomic activities into workflow activities is required to distribute work. However, the actual work is done at a much more fine-grained level. (In Figure 2, it is not possible to look into activity A. From the viewpoint of the workflow management system it is a black box.)
- Routing is used for both work *distribution* and *authorization*. As a result, only crude mechanisms can be used to align workflow and organization. (In Figure 2, cases waiting for activity B are only visible by workers with role Y. Moreover, the work is offered to all workers having this role. Therefore, it is not possible to differentiate between work that is offered and work that is visible.)
- By focusing on control flow the context, i.e., data related to the entire case and not just the activity, is moved to be background. Typically, such *context tunneling* results in errors and inefficiencies. (When executing activity B for a given case, the focus is on activity B rather than the case itself.)
- Routing focuses on what *should* be done instead of what *can* be done. This push-oriented perspective results in rigid inflexible workflows. (A workflow management system supporting the process shown in Figure 2 enforces the predefined order of activities.)

These problems are highly relevant for construction processes. Therefore, we introduce the case-handling paradigm.

The central concept for case handling is the *case* and not the activities or the routing from one in-tray to another. The case is the ‘product’ which is manufactured and at any time workers should be aware of this context. Examples of cases are the evaluation of a job application, the verdict on a traffic violation, the outcome of a tax assessment, and the ruling for an insurance claim.
To handle a case, *activities* need to be executed. Activities are logical units of work. Many workflow management systems impose the so-called ACID properties on activities [1,16]. This means that an activity is considered to be atomic and either carried out completely or not at all. We use a less rigid notion. Activities are simply chunks of work which are recognized by workers, e.g. like filling out an electronic form. As a rule-of-thumb, activities are separated by points where a transfer of work from one worker to another is likely/possible. Please note that activities separated by points of ‘work transfer’ can be non-atomic, e.g., the activity ‘book business trip’ may include activities such as ‘book flight’, ‘book hotel’, etc.
Clearly activities are related and cases follow typical patterns [5]. A *process* is the ‘recipe’ for handling cases of a given type. In many workflow management systems, the specification of a process fixes the routing of cases along activities and workers have hardly any insight in the whole. As a result

exceptions are difficult to handle because they require unparalleled deviations from the standard recipe. Since “exceptions are the rule”, precedence relations among activities should be minimized.

If the workflow is not exclusively driven by precedence relations among activities and activities are not considered to be atomic, then another means is needed to support the handling of cases. Workers will have more freedom but need to be aware of the whole case. Moreover, the case should be considered as a ‘product’ with structure and a current state. For knowledge-intensive processes, the state and structure of any case is based on a collection of *data objects*. A data object is a piece of information which is present or not present, and when it is present it has a value. In contrast to existing workflow management systems, the logistical state of the case is not determined by the control-flow status but by the presence of data objects. This is truly a shift: case handling is also driven by data-flow instead of exclusively by control-flow. This provides a balance between the data-oriented approaches of the 80-ties and the process-oriented approaches of the 90-ties.

It is important that workers have insight in the whole case when they are executing activities. Therefore, all relevant data should be presented to the worker. Moreover, workers should be able to look at other data objects associated to the case they are working on (assuming proper authorization). *Forms* are used to present different views on the data objects associated to a given case. Activities can be linked to a form to present the data objects most relevant.

Forms are only a way of presenting data objects. The link between data objects, activities, and processes is specified directly. Each data object is linked to the process. So-called *free* data objects can be changed while the case is being handled. A data object that is explicitly linked to an activity is either *mandatory* or *restricted*. If a data object is mandatory, it is required to complete the activity. If a data object is restricted, then it is required to complete the activity and it cannot be entered in preceding or subsequent activities. That means that the information can be processed if and only if at least one of the activities for which the information is restricted is now at hand.

Actors are the workers executing activities and are grouped into *roles*. Roles are specific for processes, i.e., there can be multiple roles named ‘manager’ as long as they are linked to different processes. One actor can have multiple roles and roles may have multiple actors. Roles can be linked together through role graphs. A role graph specifies ‘is_a’ relations between roles. This way, one can specify that anybody with role ‘manager’ also has the role ‘employee’.

For each process and each activity three roles need to be specified: the execute role, the redo role, and the skip role.

- The *execute role* is the role that is necessary to carry out the activity or to start a process.
- The *redo role* is necessary to undo activities, i.e., the case returns to the state before executing the activity. Note that it is only possible to undo an activity if all following activities are undone as well.
- The *skip role* is necessary to pass over activities. In order to skip over two consecutive activities, the worker needs to have the skip role for both activities.

The three types of roles associated to activities and processes provide a very powerful mechanism for modeling a wide range of exceptions. The redo ensures a very dynamic (as it is dependent on the role of the employee and the status of the case) and flexible form of a *loop*. The skip takes care of a range of exceptions that would otherwise have to be modeled in order to pass over activities. Of course, there are ways of avoiding undesirable effects: you can define the ‘*no-one’* or ‘*nobody’* role in every process that is higher than all the other roles and that no user can perform. You can also define an ‘*everyone*’ role that is lower than all others. An activity with the ‘*no-one’* redo role can never be undone again and it would then also not be possible to go back to an earlier activity. This is a very effective way to model ‘points of no return’. An execute *everyone* role means that the activity can be carried out by anyone who at least has a role in that process (because that person is then, after all, at least equal to the *everyone* role). Note that in addition to these three roles, one could consider additional roles, e.g., the “responsible role” or the “supervisor role”. For a case one could also define the “case manager role”, etc. The variety of roles associated to a case or an activity shows that in case handling it is possible to separate authorization from work distribution. When using the classical in-tray, one can only see the work-items, which need to be executed. The only way to get to a case is through work-items in the in-tray, i.e., authorization and work distribution coincide. For case handling the in-tray is replaced by a flexible *query mechanism*. This mechanism allows a worker to navigate through all active cases. The query “Select all cases for which there is an activity enabled which has an execute role R” can be used to emulate the traditional in-tray. In fact, this query corresponds precisely to the work queue concept used in the in-tray of the workflow management system Staffware. By extending the query to all roles a specific worker can fulfill, it is possible to create a list of all cases for which the worker can execute activities at a given point in time. However, it is also possible to have queries such as "Select all cases that worker W worked on in the last two months" and "Select all cases involving more that 80k Euro for which activity A is enabled". By using the query mechanism workers can get a handle to cases that require attention. Note that authorization is separated from work distribution. Roles are used to specify authorization. Standard queries can be used to distribute work. However, the query mechanism can also be used to formulate ad-hoc queries, which transcend the classical in-tray.

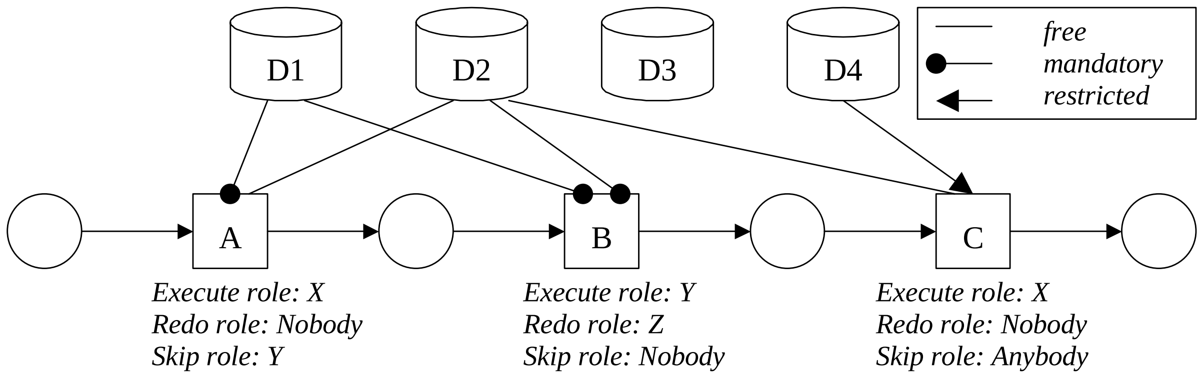

Figure 3: an example illustrating some of the case-handling concepts

To illustrate some of the concepts associated to case handling we revisit the example of Figure 2. In Figure 3 some of the concepts have been added graphically. Note that each activity has three roles associated to it. For example, activity A may be executed by a worker having role X. However, it can also be skipped by a worker having role Y. Assuming that there is no worker having role Nobody, this activity cannot be redone. Similarly, activity B cannot be skipped but can be redone by a worker having role Z. Figure 3 also shows the free, mandatory, and restricted data objects associated to each activity. Activity A requires data element D1 to be set because D1 is mandatory for the completion of A. Activity B has two mandatory data objects (D1 and D2). D4 is restricted to C. Note that because of the explicit modeling of data objects, activities are no longer "black boxes" and their interrelationships become clear. Moreover, it is possible to remove explicit causal relationships and it is possible to execute several activities in one activity. Consider for example the scenario where a worker enters both D1 and D2 when executing activity A. After completing A all mandatory data elements for B are present which indicates that B is not required. Using this information B is skipped and C is the next activity to be executed.

Table 1: differences between Workflow Management and Case Handling

	Workflow management	**Case handling**
Focus	*Work-item*	*Whole case*
Primary driver	*Control flow*	*Case data*
Separation of case data and process control	*Yes*	*No*
Separation of authorization and distribution	*No*	*Yes*
Types of roles associated with activities	*Execute*	*Execute, skip, redo, …*

To conclude this section, we summarize the main differences between workflow management, as supported by contemporary workflow technology, and case handling (see Table 1). The focus of case handling is on the whole case, i.e., there is no context tunneling by limiting the view to single work-items. The primary driver to determine which activities are enabled is the state of the case (i.e., the case data) and not control-flow related information such as the activities that have been executed. The basic assumption driving most workflow management systems is a strict separation between data and process. Only the control data is managed. The strict separation between case data and process control simplifies things but also creates integration problems. For case handling the logistical state of a case (i.e., which activities are enabled) is derived from the data objects present, therefore data and process cannot be separated! Unlike workflow management, case handling allows for a separation of authorization and distribution. Moreover, it is possible to distinguish various types roles, i.e., the mapping of activities to workers is not limited to the execute role.

3. Case handling in the building industry

In the previous section, the case-handling paradigm was explained by comparing case handling with traditional workflow management systems. At this point it should be mentioned that there are also so-called ad-hoc workflow management systems. Examples of such systems are InConcert (TIBCO), Ensemble (Filenet), and TeamWARE Flow (TeamWARE Group). These systems allow for the creation and modification of workflow processes during the execution of the processes. Each case has a private process model and therefore the traditional problems encountered when changing a workflow specification can be avoided. Ad-hoc workflow management systems allow for a lot of flexibility. The workflow management system InConcert even allows the user to initiate a case having an empty process model. When the case is handled, the workflow model is extended to reflect the work conducted. Another possibility is to start using a template. The moment a case is initiated, the corresponding process model is instantiated using a template. After instantiation, the case has a private copy of the template, which can be modified while the process is running. InConcert also supports “workflow design by discovery”: The routing of any completed workflow instance can be used to create a new template. This way actual workflow executions can be used to create workflow process definitions.

Thus far we have considered three types of systems: (1) traditional (or production) workflow management systems (e.g., Staffware), (2) ad-hoc workflow management systems (e.g., InConcert), and (3) case-handling systems (e.g., FLOWer). Before, we discuss the suitability of these three types of systems in construction, we use Figure 4 to position case handling. Case-handling systems are both data and process driven and target at structured or implicitly structured processes. Case-handling systems try to balance data and processes. The processes supported can be either completely structured (i.e., all possible routes are explicitly specified) or variable in the sense that implicit deviations of specified routes are possible. If for each activity the skip and redo roles are assigned to the ‘nobody’ role (i.e., no-one can skip or redo activities), the process supported by the case handling tool is completely structured. By allowing skip and redo roles to be assigned to other roles, implicit deviations of the specified route are possible. Production workflow (i.e., the traditional workflow management systems) is process driven (data is only introduced for routing reasons) and supports only structured processes. Note that structured processes may allow for many alternative routes. The criterion is whether these routes are fixed at design time and specified explicitly, or not. Ad-hoc workflow is also process driven but aims at variable processes that can be changed on the job. Note that these systems support a different type of variability: At any point in time the process is fixed and explicitly specified, however, end-users may change the workflow specification on the job. For reasons of completeness, Figure 4 also shows a characterization of the class of groupware systems (e.g., Lotus Notes). These systems are data driven and support unstructured processes, i.e., there is no notion of an explicit process model.

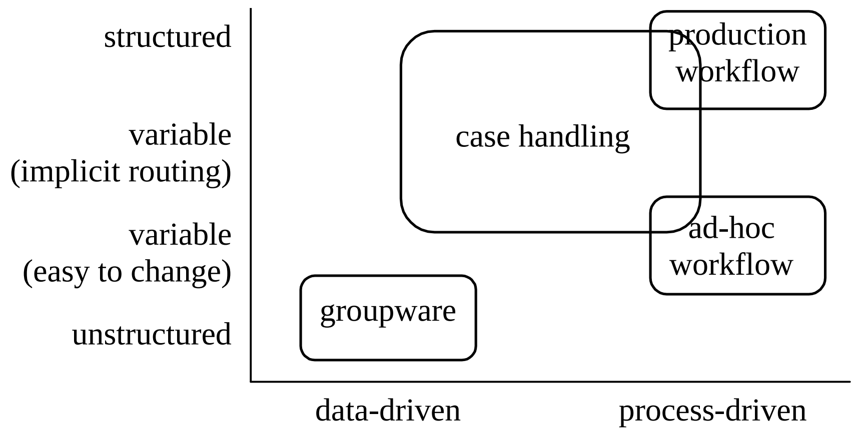

Figure 4: positioning case handling

Using Figure 4 we can think about the suitability of the various systems to support construction processes. Production control in construction is quite different from production control in other parts of the industry. Concerning workflow control, the following differences are relevant:

- Within one construction company different projects are carried out. Production control in construction depends on the complexity of the companies projects [21,25]. Characteristics as the amount of different resources (as well as internal as external), the need for activity related information and the volume of it, as well as the extent of the project determine the complexity of projects. This means that the characteristics of the construction processes within a specific construction company differ from project to project. A workflow management system which supports these processes has to be flexible in such a way that the user (e.g. the project manager or the work planner) can choose certain paths within the work flow system [25]. He only uses the (prescribed) paths, which are relevant to the ongoing project. Within the company templates of possible flow-paths have to be developed. During the execution of the project the user has to choose the template which fits the best. In terms of Figure 4 we can speak about variable structured processes (implicit routing).

- The existence of a high amount of uncertainty within the process is a very important characteristic of construction projects. At the beginning of a project a lot of uncertainty exists about the specifications of the construction object to be made. In a lot of cases specific design variables are not known. This means that during the execution of the project the implications of the (dynamic) design process are not yet available. The need for specific external resources (subcontractors, installers, etc.) and what kind of activities have to be carried out by these resources is only known during the project. Also the progress of processes within the project, e.g. the availability of materials and weather influences are all unknown quantities. In other words, in construction the availability of information with respect to the specifications or the progress of the construction object plays a crucial role in process control. The process model described in Section 4 (Figure 5 and Figure 6) clearly illustrates this. As a result of this, the state of a project is mainly determined by the presence of data objects such as drawings and progress data and not so much by predefined routings of activities. To cope with this uncertainty, a supporting workflow management system has to be very flexible. Workers must be able to make choices about the sequence of certain activities. Also making choices of skipping or re-doing activities [12].

Summarizing, construction processes are dependent on the specific needs of the project, which can vary from project to project. However it is possible to classify projects and describe variable structured processes (with implicit routing). The presence of a lot of uncertainty means that processes have to be flexible, changeable. Figure 4 shows that the traditional production workflow systems will not support these kinds of projects, i.e., traditional workflow systems are eliminated as a potential candidate because of the lack of supporting variable, unstructured processes (see Table 1).

The fact that activities in construction are often triggered by the presence of new data objects means that the user of such a system has to be able to oversee the whole project and not only a part, a work item, of the project. The link that exists between authorization and distribution of activities, in traditional workflow management systems, is therefore in construction not desirable. In that case the user only sees the activities in his own in-tray. The own initiative of the user is then impossible, while this is an important aspect in construction on account of the needed flexibility. Case handling systems meet this flexibility even better because a user can not only execute an activity, but can also skip or redo an activity. In that way the user can introduce loops himself and can influence the process course to previously determined extent.
Also we can rule out groupware systems such as Lotus Notes for this purpose. These systems are unaware of the processes taking place and, therefore, cannot be expected to offer process support. Note that it is possible to use groupware systems to implement support for construction processes. However, in this case either the processes are hard-coded in applications or handed over to specific workflow modules (e.g., Lotus Domino Workflow). In both cases, the groupware system itself is not offering process support. Ad-hoc workflow management systems seem to be attractive for construction processes. However, there are two important requirements for being able to apply ad-hoc workflow management systems. The first requirement is that workers are aware of the processes they are dealing with. This means that the processes should only be defined or modified by workers having a good overview of the whole process. The second requirement is that workers have the ability to use advanced modeling tools and have a good understanding of process modeling techniques. It is essential that modelers can think in terms of sequential, parallel, conditional, and iterative routing. The first requirement does not inhibit the use of ad-hoc workflow management systems in construction. However, the second requirement poses serious problems. In construction, but also in many other application domains, it is not realistic to assume that workers can create and modify process models. Especially constructions which involve a mixture of choice and synchronization (i.e., conditional and parallel routing) are difficult to handle by end-users. Another problem of using ad-hoc workflow management systems is the fact that it is difficult to control the processes: Too much flexibility may have undesirable effects if it is misused.

Given the limited applicability of traditional groupware and ad-hoc workflow management systems, only one type of system remains: case-handling systems. Therefore, we advocate the use of case handling.

4. Heijmans case: Process model

To find out whether case handling is applicable in construction processes, a process model was built. The goal of this model was to give better insight in the process concerned and in the feasibility of case handling systems in construction. The process model addresses three essential issues:

- *Flexibility*
  How to model project specific aspects (as known in construction)?
- *Standardization*
  Can the process concerned be standardized such that it can be automated and it is not necessary to change the process model for each project?
- *Level of detail*
  What is the level of detail in defining the activities in the process model? If the process model is very detailed, the model is hard to adjust to several projects. If the model abstracts from essential issues, the model is too general and the automation based on the model cannot offer the requested support in managing.

As mentioned in the introduction, the process model contains the processes of preparing HVAC and of preparing semi-prefab concrete floor elements. Besides these two processes the process model also contains subprocesses from the Preparation of Execution Phase (PEP) that are necessary to execute the two preparation processes. Figure 5 shows an overview of these processes. The goal of this figure is not to provide details, but to illustrate the complexity of the overall process. As Figure 5 shows, the overall process consists of the following subprocesses:

- The process of filling in the manual, this includes the creation, check, and approval of the needed schedules and documents for the manual;
- The process of drawing by the architect, this includes the transformation of specification drawings to work drawings by the architect;
- The process of drawing by the structural engineer, this includes the adaptation of specification drawings to work drawings by the structural engineer;
- The process of purchasing, this includes the division of activities concerning purchasing until the concluding of contracts.

The process model is set up in PROTOS, which uses the Petri-net technique for modeling workflow processes [1]. PROTOS is a product of Pallas Athena, a Dutch software supplier. The process model starts with handing over the project from the Project Development Phase (PDP) to the Preparation of Execution Phase (PEP). The last activity in the process is handing over the project from the Preparation of Execution Phase to the Realization of Execution Phase (REP). The process model contains 170 activities and 300 data documents. The required flexibility is partly handled by introducing loops and conditional routing.

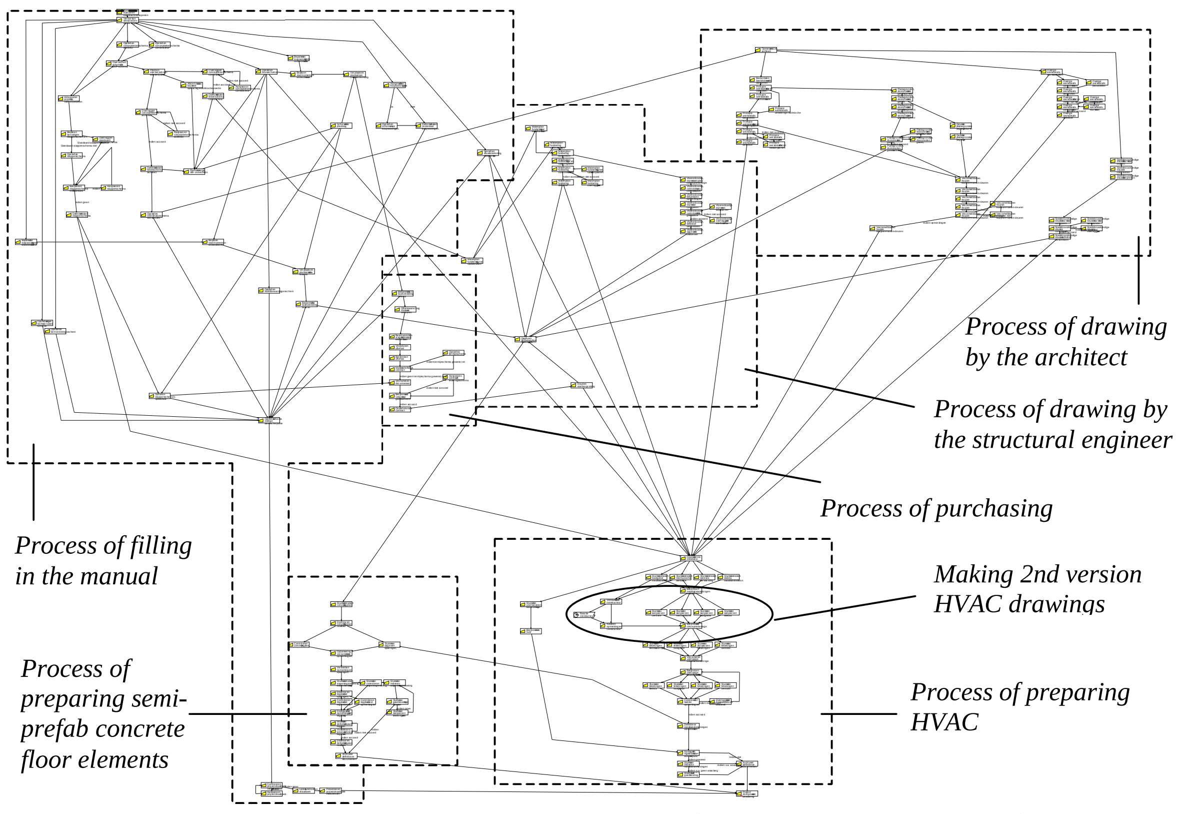

Figure 5: overview illustrating the various parts of the HVAC and semi-prefab concrete floor elements preparation processes

The process of making the second version of HVAC drawings (Figure 6) is discussed in more detail. In Figure 6 the joins and splits for the activities are characterized by a white (OR) and a grey (AND) circle-part. An activity having an OR-Join only needs one of the preceding activities to be completed, while an activity having an AND-Join needs all preceding activities to be completed. The same idea works for the exit of an activity. An OR-Split only enables one of the subsequent activities, while an AND-Split enables all the subsequent activities. The activities in Figure 6 with no circle-part only have one arrow incoming or outgoing. The two places in the model have an OR-Join and an OR-Split (indicated by a circle).

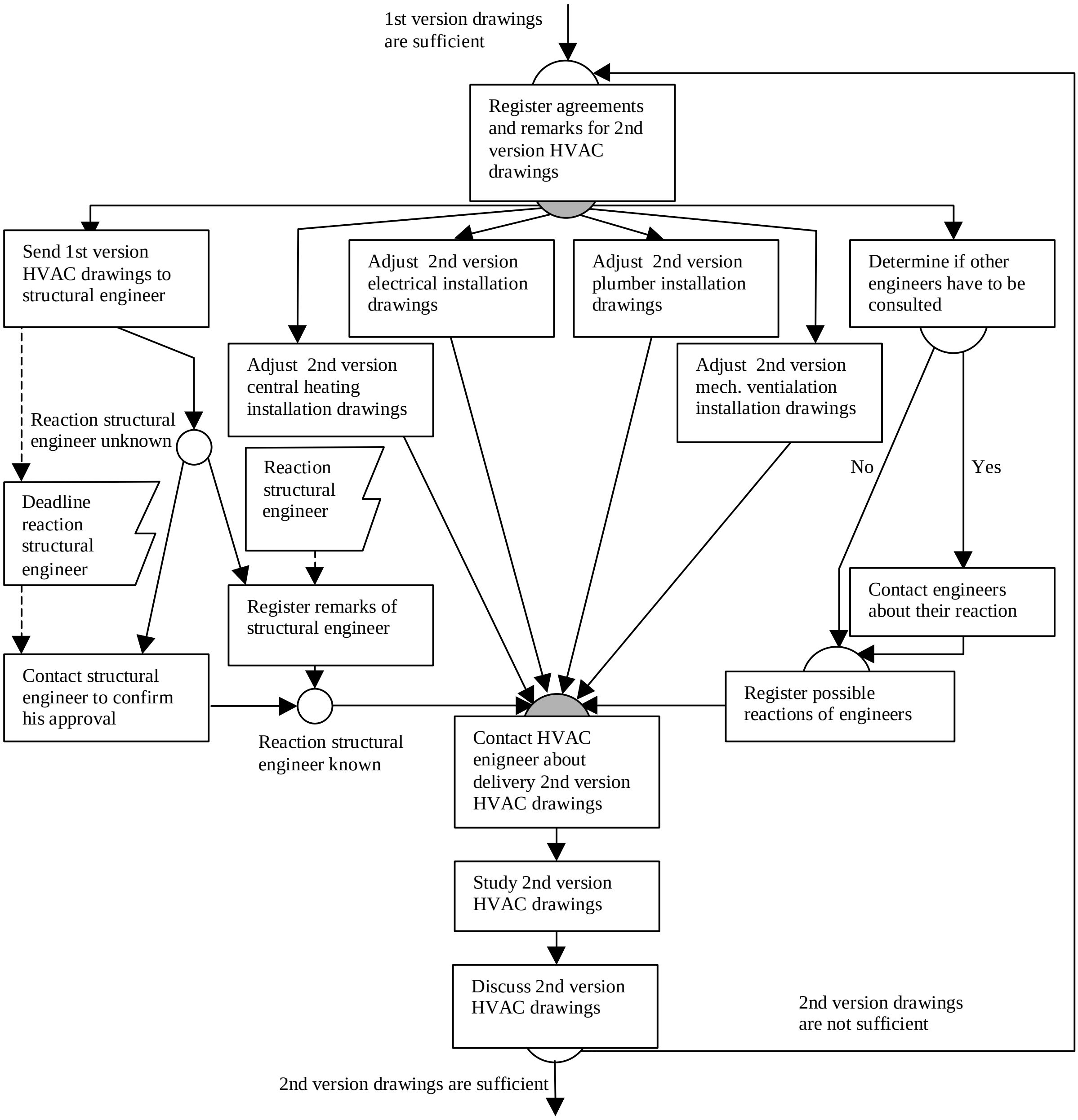

Figure 6: making of the second version HVAC drawings

After having approved the first version of HVAC drawings the first activity in the model (Register agreements and remarks for 2nd version HVAC drawings) is enabled. Completing this activity is followed by six parallel activities. After sending the first version of HVAC drawings to the structural engineer two conditional routings are defined, determined by the reaction of the structural engineer. This reaction is translated in two triggers, “Deadline for reaction structural engineer” and “Reaction structural engineer”. If the deadline expires, the activity of “Contact structural engineer to confirm his approval” is enabled. If the structural engineer reacts before passing the deadline, “Register remarks of structural engineer” is enabled. At the right side of the model the user determines the routing to follow. If any other engineers have to be consulted, the user chooses “Yes” and enables “Contact engineers about their reaction”. By choosing “No” the user enables directly “Register possible reactions of engineers”. Completing all preceding, parallel activities enables “Contact HVAC engineer about delivery 2nd version HVAC drawings”. Having received the drawings the user has to study the drawings before he is able to discuss the drawings. In the discussion the decision is made whether the

2nd version HVAC drawings are sufficient or not. If the drawings are not sufficient the process goes back to the first activity, “Register agreements and remarks for 2nd version HVAC drawings” (not in this figure). Otherwise the process continues with delivery of the provisional HVAC drawings. For each activity in the model the information summarized in Table 2 has been defined.

Table 2: information about an activity

Basis	Adjust 2nd version central heating installation drawings
Role	Work planner
Description	Manage the HVAC engineer in adjusting the central heating installation drawings
Instruction	Contact the HVAC engineer frequently to check if he has any questions or remarks about adjusting the drawings. Tell him the deadline for delivering the drawings and ask him whether he can meet the deadline.
Responsible	Project manager
Team	Section Preparation
Data	
Created	Second version of central heating installation drawings (document, paper, internal)
Optional	Contract, what has been stated about the second version of central heating installation drawings (document, paper, internal)
	First version of central heating installation drawings (document, paper, internal)
	Deadline for second version of central heating installation drawings (document, paper, internal)
	Remarks and questions of the HVAC engineer during the making of the first version of central heating installation drawings (document, paper, internal)
	Remarks, questions, and agreements from the discussion about the first version of HVAC drawings (document, paper, internal)
	Activity schedule for preparation process HVAC (document, paper, internal)
	Number of version of the saved second version of the central heating installation drawings (document, paper, internal)
	Own remarks and question in making the first version of central heating installation drawings (document, paper, internal)
	Agreed and stated layout HVAC drawings (document, paper, internal)
Incoming connections	Register agreements and remarks for 2nd version HVAC drawings
Outgoing connections	Contact HVAC engineer about delivery of 2nd version HVAC drawings
Concerned parties	HVAC engineer
Entrance	AND-join (only one activity)
Exit	AND-split (only one activity)

The process model answers the three questions, stated at the beginning of this paragraph:

- Flexibility: Most of the project aspects can be modeled by introducing conditional routing and loops, to meet all project aspects costs a lot of effort while it is probably unnecessary;
- Standardization: The project manual of Heijmans Bouw contains standard schedules to execute parts of the complete process. These schedules show standardized processes that can be translated into a model;
- Level of detail: The level of detail in defining the activities is comparable to the level used in the project manuals. Experience with the manual learned that this level is suitable for use in several projects. Therefore, a system based on these activities can be expected to be adjustable to several projects without being too determining towards the users.

On basis of the process model we decided to continue the research by developing a prototype. This prototype can give better insight in the possibilities ("proof-of-concept") and enables the discussion on the possibilities and feasibility of case handling in construction [26].

5. Heijmans case: Implementation using FLOWer

The prototype is defined according to the process model that represents the present way of working by Heijmans Bouw. FLOWer, the case handling system of Pallas Athena, is used to build the prototype [22]. We did not specifically chose for FLOWer; the main focus of this research is the principles of case handling and not one specific system.

FLOWer Studio is the design tool of FLOWer and is used by the system developer and the system administrator. In FLOWer Studio the roles and activities in the processes concerned are defined (Figure 7 and Figure 8). The four roles defined in the prototype are 'Nobody', 'Assistant-director', 'Project-manager', and 'Work planner'. The hierarchy of the roles determines the responsibilities in the process. Each activity from the process model has to be translated into electronic forms that the user will see in the case handling system. The process model describes all the activities that have to be done while FLOWer describes the transfer of work. Therefore a form can cover more activities in the process model. The loops in the process model are translated in FLOWer by assessing the redo-activity to one of the roles. For each activity this information can be defined in the forms showed by Figure 7 and Figure 8.

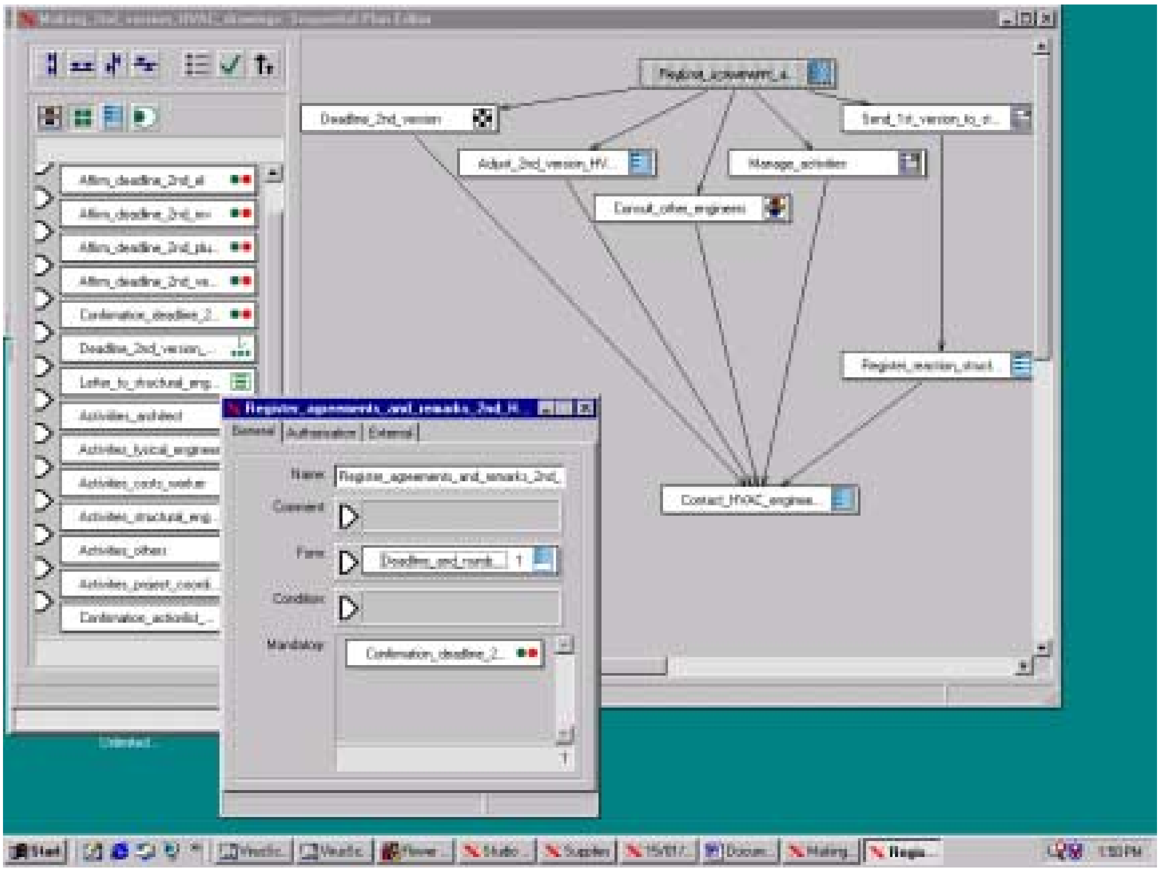

Figure 7: defining activities

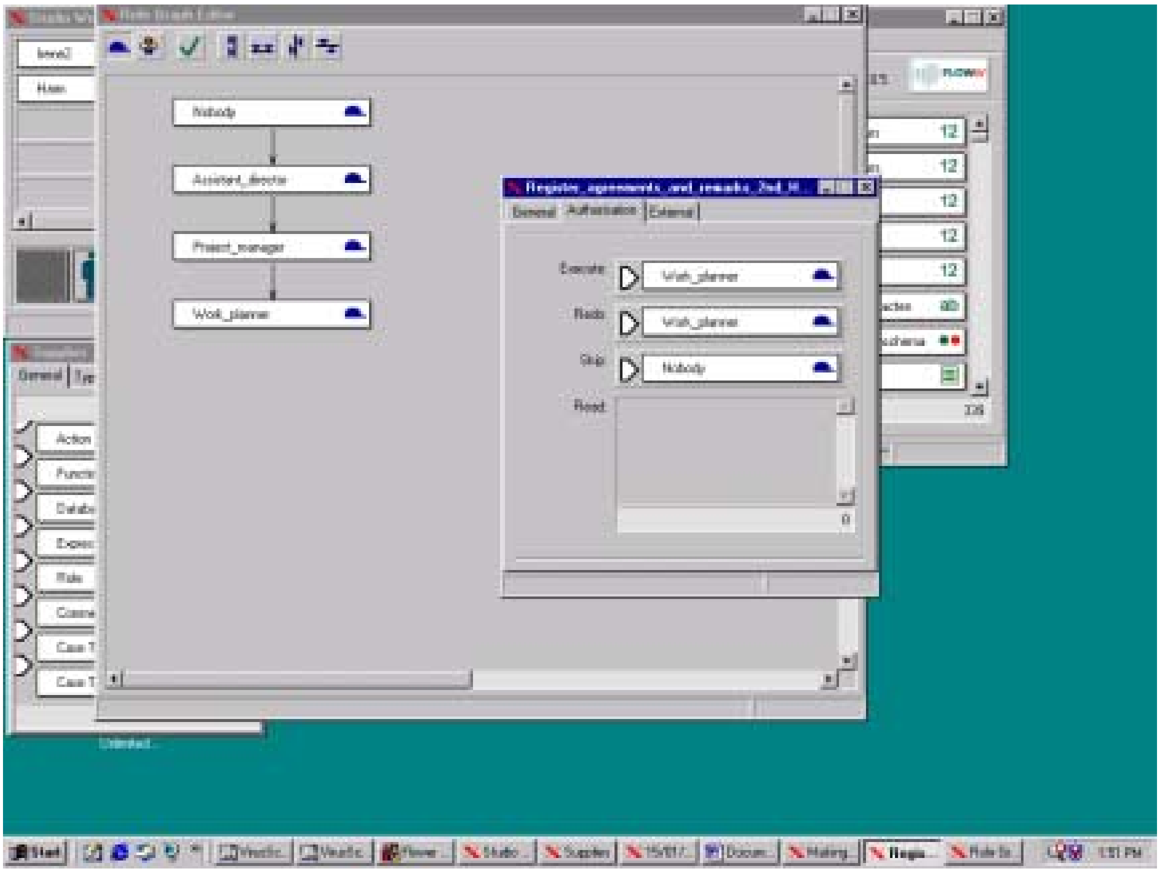

Figure 8: defining roles

A user can start a new case or open an existing case in FLOWer Case Query. Figure 9 shows an example of a user form (left) concerning the making of the second version HVAC drawings. The right side of the screen provides an overview of the project. The user sees the activities that are already carried out or skipped. These activities are displayed at the right side of the status line. Activities on the left side of this line cannot be executed because previous activities need to be finished first. Activities on the line can be executed. The form shown at the left side of the screen corresponds to the activity that is highlighted on the status line. A user can only see forms that (s)he is allowed to execute. Data fields marked by a rectangle in front of the field are mandatory.

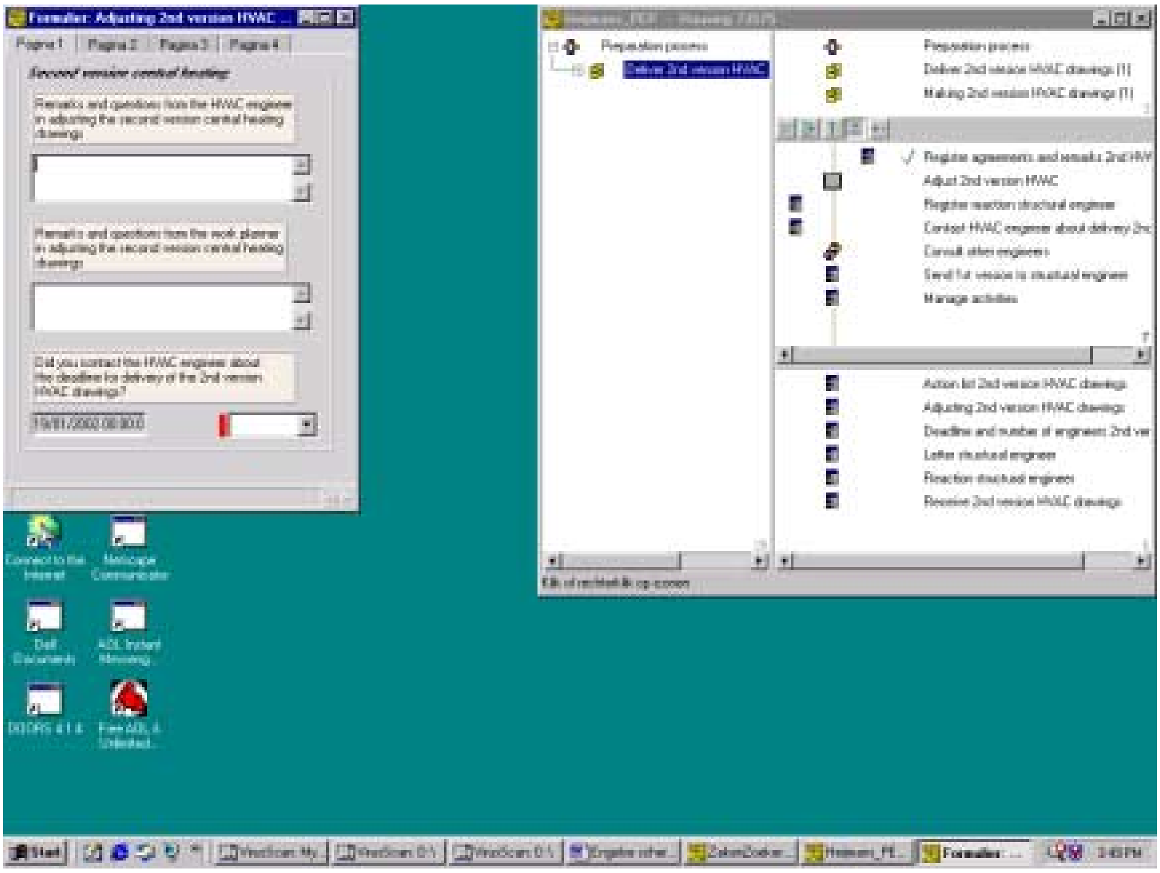

Figure 9: screenshot of the prototype showing an overview of the status of the case (right) and a form (left)

A special kind of form is the Standard Letter Action. This form can generate documents with information filled in earlier in the project. Figure 10 shows the example of generating a letter to the structural engineer in Microsoft Word.

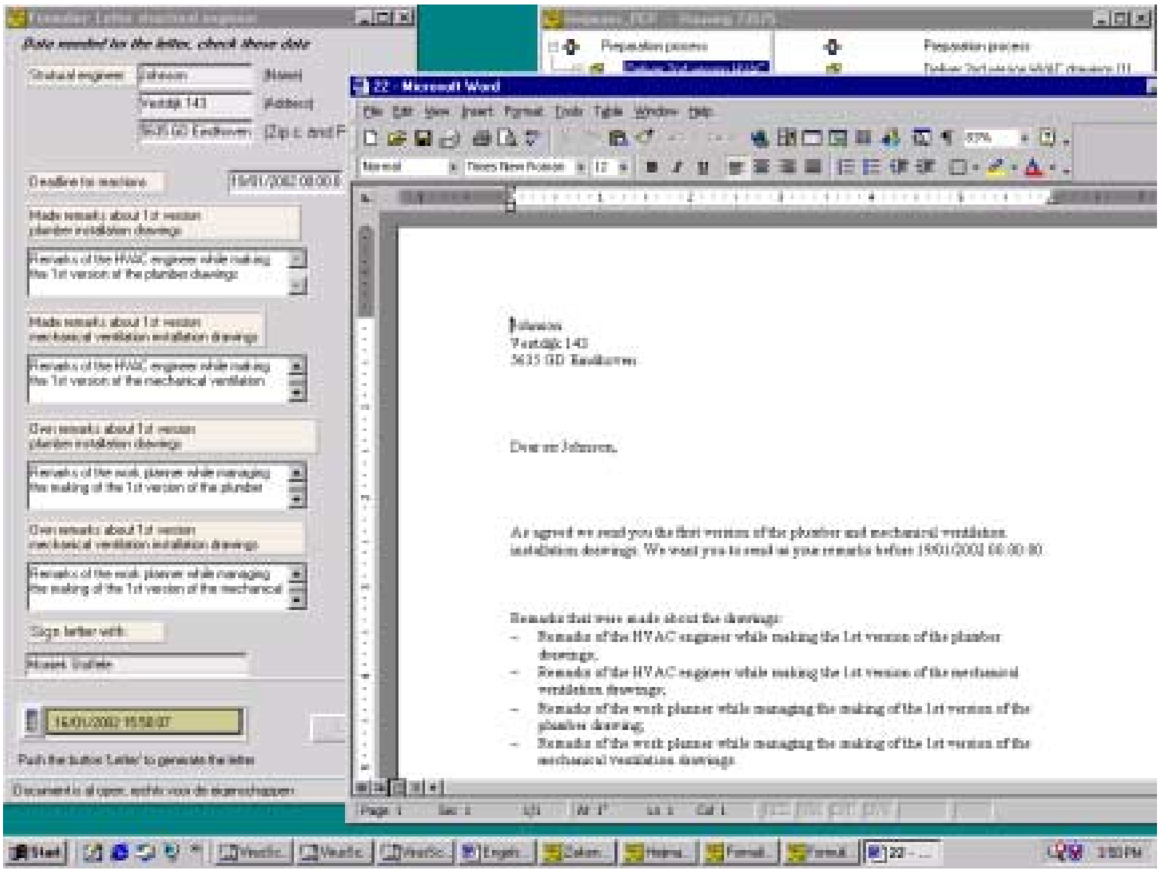

Figure 10: generating a letter in Word

6. Heijmans case: Evaluation

Three work planners of Heijmans Bouw tested the prototype. During the test the workers were observed and questioned. Afterwards more questions were posed to them. The test focused on five aspects:

- Practical feasibility: Does the prototype meet the level of knowledge, the knowledge of computers, and the present way of working of the users?
- Economical feasibility: Do the benefits in development, implementation, and use of the prototype exceed the costs of development, implementation, and use?
- Flexibility: Does the prototype offer enough flexibility so that it can meet project specific aspects to a certain extent?
- Support: To what extent does the user expect the prototype to support him in managing the project?
- Further development: Is the prototype interesting enough for further development? If so, what are the points of attention for that development?

The reactions of the work planners were very positive. The main conclusions concerning the aspects of the testing are discussed in Table 3.

Table 3: evaluation results

Practical feasibility	The prototype meets the level of knowledge, the knowledge of computers, and the present way of working of the users. The prototype supports mainly the information handling in the process. The prototype needs to produce better overview of the process, especially for tracing back information more rapidly and easily. The overview can also be used to show the critical path in the project.
Economical feasibility	The prototype has benefits in quality and time. The benefits would be greater if the prototype gave a good overview of the process, enabling rapid information tracing and saving time.
Flexibility	The flexibility of the prototype is comparable to that of the manuals. Important point of attention is how the system can meet project specific aspects that are not in the system. In the development a choice has to be made whether the system is only suited for private housing projects or also for business housing projects.
Support	The prototype is a valuable support especially for the inexperienced work planner and in complex projects. The making of an action list is more attractive if it contains the whole process and not, like in the prototype, only for a part of the process.
Further development	The development has to start simple and small. Using prototypes in pilot-projects has to lead to the introduction of the system. During the introduction are courses for the later user very important. Clear agreements have to be made for the use of the system, is the system obligatory or free to use. Involving external parties is not interesting (at this moment). From the experiences of the workers some points of attention for the system are defined.

The prototype provides a proof-of-concept that demonstrates that case handling is an interesting technology for construction companies like Heijmans Bouw. At this point in time, we cannot provide empirical evidence that case handling economically feasible. However, we can conduct a “Gedanken Experiment’’ to estimate potential benefits. Based on our experiences, we estimate that it would take four person-years to model the overall preparation processes of Heijmans, to implement this using FLOWer, and to configure the whole system. This estimation is based on the time spent on the process model and prototype system. (Note that the prototype is fully operational!) Taking development costs of 70,000 Euro per year, the development costs are 240,000 Euro. Typical costs for software and training when introducing workflow technology are 1000 Euro per worker. Within Heijmans there are about 40 work planners. Therefore, the software and training costs would amount to 40,000 Euro. This brings the total costs to 280,000 Euro. It is difficult to estimate the benefits. Based on our experiences and taking into account the comments of the work planners that worked with the system, we estimate the benefits to be a reduction of 10 to 20 percent in working time of the work planners. This is supported by other studies indicating that professionals spend 18 percent of their time looking for the right information [9]. The salary costs of a work planner are about 50,000 Euro per year. If it is possible to save 10 percent of labor costs, the saving are 0.10 * 40 * 50,000 = 200,000 Euro. If the savings are 20 percent, this increases to 400,000 Euro per year. This implies that the Return On Investment (ROI) is in-between 0.7 and 1.4 years. Please note that we cannot validate these figures using empirical data. However, if our assumptions are correct, the introduction of case handling is definitely economically feasible. Therefore, Heijmans Bouw is very interested in the outcome of this research and is investigating the application of workflow and case handling technology in a broader context.

7. Conclusions

This paper is focused on the use of workflow management systems in construction. The successfulness of using workflow management systems in other industries was the reason for developing and carrying out the research described in this paper. In fact, two different worlds came together in this research: on the one hand the world of ongoing, scientific research on the use of workflow management systems in parts of the industry in which well structured, data related processes take place, like in assurance companies and banking. On the other hand the world of construction, which can be described as unstructured processes involving a lot of uncertainty.

Bringing these two worlds together lead to a two forked approach to the research:

- Can the concept of workflow management used in the production industry be suitable for unstructured processes? As a result of the specific characteristics as described in this paper, the construction industry is a very interesting part of the industry to investigate this research question.
- Will workflow management improve the control of processes in the construction industry?

This paper shows that both questions can be answered positively to a certain extent. It can be concluded that also unstructured processes like the construction industry can be modeled, on condition that the user can make choices about the relevance of certain routings and besides this can skip or redo activities. This means that application of workflow management in the industry will be interesting for more companies than the typical "administrative factories" (e.g., insurance, banking and government) described in literature [20]. This makes the application of workflow management to the loosely structured processes of Heijmans highly relevant. Carrying out research on and developing systems that cope with these kinds of unstructured processes has to be a main objective for the forth-coming years.

The research described in this paper also shows that the construction industry, which is an industry with a high amount of failure costs can be improved by using a specific form of workflow management, namely case handling. However, using this kind of system means a big change from the current way of working within the construction industry. Developing this concept is a great opportunity for the construction industry. Therefore, case handling has to be placed on the research agenda to further investigate the possibilities, e.g. the specific impact on processes and people within the construction industry needs to be investigated in more detail.

References

1. W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21-66, 1998.
2. W.M.P. van der Aalst. On the Automatic Generation of Workflow Processes Based on Product Structures. *Computers in Industry*, 39:97-111, 1999.
3. W.M.P. van der Aalst and P. Berens. Beyond Workflow Management: Product-driven Case Handling. In S. Ellis and T. Rodden and I. Zigurs (Editors): *International ACM SIGGROUP Conference on Supporting Group Work (GROUP 2001)*, pages 42-51, ACM Press, New York, 2001.
4. W.M.P. van der Aalst, J. Desel, and A. Oberweis, editors. *Business Process Management: Models, Techniques, and Empirical Studies*, volume 1806 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2000.
5. W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Advanced Workflow Patterns. In O. Etzion and P. Scheuermann, editors, *7th International Conference on Cooperative Information Systems (CoopIS 2000)*, volume 1901 of *Lecture Notes in Computer Science*, pages 18-29. Springer-Verlag, Berlin, 2000.
6. W.M.P. van der Aalst and S. Jablonski, editors. *Flexible Workflow Technology Driving the Networked Economy*, Special Issue of the *International Journal of Computer Systems, Science, and Engineering*, volume 15, number 5, 2000.
7. A. Agostini and G. De Michelis. Improving Flexibility of Workflow management Systems. In [4], pages 218-234.

8. C.J. Anumba and R. Ruikar, Electronic Commerce in construction: Trends and Prospects, *Automation in Construction*, volume 11, pages 265-275, 2002.
9. C. Bots. On the Effectiveness of PDM systems within Stork. Master Thesis, Eindhoven University of Technology. 1996.
10. P. Brandon, M. Betts and J.W.F. Wamelink, Information Technology Support to Construction Design and Production, *Computers in Industry*, Special issue IT in construction, Volume 35, pages 1-12, 1998
11. Casati, S. Ceri, B. Pernici, and G. Pozzi. Workflow Evolution. *Data and Knowledge Engineering*, 24(3):211-238, 1998.
12. M. Crow and S. Kydd, Agents and Suggestions in a Web-based Dynamic Workflow Model, *Automation in Construction*, volume 10, pages 639-643, 2001
13. C.A. Ellis and K. Keddara. A Workflow Change Is a Workflow. In [4], pages 218-234.
14. M. Hammer and J. Champy. *Reengineering the corporation*. Nicolas Brealey Publishing, London, 1993.
15. T. Herrmann, M. Hoffmann, K.U. Loser, and K. Moysich. Semistructured models are surprisingly useful for user-centered design. In Coop2000, Sophia-Antipolis, France, May 2000.
16. S. Jablonski and C. Bussler. *Workflow Management: Modeling Concepts, Architecture, and Implementation* International Thomson Computer Press, 1996.
17. M. Klein, C. Dellarocas, and A. Bernstein, editors. *Proceedings of the CSCW-98 Workshop Towards Adaptive Workflow Systems*, Seattle, Nov. 1998.
18. M. Klein, C. Dellarocas, and A. Bernstein, editors. *Adaptive Workflow Systems*, special issue of the journal of *Computer Supported Cooperative Work*, volume 9, numbers 3-4, 2000.
19. L. Kornelius and J.W.F. Wamelink, The Virtual Corporation, Learning from Construction, *Supply Chain Management*, volume 3, number 4, pages 193-202, 1998
20. P. Lawrence, editor. *Workflow Handbook 1997, Workflow Management Coalition*. John Wiley and Sons, New York, 1997.
21. B. Melles and J.W.F. Wamelink, *Production Control in Construction, Different Approaches to control, use of information and automated data processing*, Dissertation; Delft University of Technology; Delft University Press, The Netherlands, 1993
22. Pallas Athena. *Case Handling with FLOWer: Beyond Workflow*. Internal report, Pallas Athena, Apeldoorn, The Netherlands, 2000.
23. Staffware. *Staffware Case Handler*. White paper, Staffware PLC, Berkshire, United Kingdom, 2000.
24. D.M. Strong and S.M. Miller. Exceptions and exception handling in computerized information processes. *ACM Transactions on Information Systems*, 13(2):206-233, 1995.
25. Z. Turk and B. Lundgren, Communication Workflow Perspective on Engineering Work, *Concurrent Engineering in Construction, Challenges for the new millennium*, CIB publication 236, editors: M. Hannus, M. Salonen, A. Samad Kazi, Espoo, 1999.
26. J.W.F. Wamelink, M. Stoffele, and W.M.P. van der Aalst. Workflow Management in Construction: Opportunities for the Future. In K. Agger, P. Christiansson, and R. Howard, editors, *Proceedings of CIB W78 conference 2002: Distributing Knowledge in Building*, pages 115-122, Aarhus, Denmark, 2002. International Council for Research and Innovation in Building and Construction.
27. M. Weske. *Formal Foundation and Conceptual Design of Dynamic Adaptations in a Workflow Management System.* In Sprague (Editor): Proceedings of the Thirty-Fourth Annual Hawaii International Conference on System Science (HICSS-34). IEEE Computer Society Press 2001.