European Concurrent Engineering Conference 1998, Erlangen-Nuremberg, Germany, April 1998

THE WEP MODEL: ADEQUATE WORKFLOW-MANAGEMENT FOR ENGINEERING PROCESSES

Thomas Beuter*, Peter Dadam**, Peter Schneider*

* Daimler-Benz AG
Research and Technology
Postfach 2360
D-89013 Ulm, Germany
Email: Thomas.Beuter@dbag.ulm.DaimlerBenz.COM
Peter.Schneider@dbag.ulm.DaimlerBenz.COM

** University of Ulm
Dept. of Databases
and Information Systems
D-89069 Ulm, Germany
Email: dadam@informatik.uni-ulm.de

KEYWORDS

engineering process, flexible workflow management, concurrent engineering

ABSTRACT

Reducing costs of engineering processes like the *design or change of components* is a key issue in the automotive engineering area. Like in other domains, the use of process-oriented workflow management systems (WfMS) is a promising approach to achieve this goal. However, there are additional features required which todays WfMSs do not offer. These application specific requirements are outlined in more detail. Then a new WfMS, the **WEP** model (Workflow-Management for Engineering Processes), is introduced. The **WEP** model offers new concepts for modelling the features required in this application domain and provides new interaction forms at runtime. These are the integration of unstructured creative subprocesses, the release and control of preliminary data, and the support of coordination phases.

1 MOTIVATION

Similar to processes in many other areas there is an increasing demand on reducing the costs (time, money) of engineering processes like the *design or change of components*. One promising approach is, like in the area of business processes, to use process-oriented workflow management systems (WfMS) to achieve this goal. There are, however, significant differences between (administratively oriented) business processes and engineering processes. Therefore, workflow technology can only be successfully applied in this area if these differences are reflected in the WfMS technology, accordingly.

These differences are shown in a simplified engineering process, the change of components, *CoC* (see figure 1). A CoC process starts with the selection of components to be changed. Additionally, a deadline is specified up to this the new CoC process has to be finished. Then, several design engineers start with their *individual* and *creative* design activities. During such a design activity the design engineer is using different tools (e.g. CAD, simulation and documentation tools) and constructs variations of the component. At the end the design engineer selects one of the component's variants and passes it to a DMU-checker. A DMU-checker builds virtual prototypes (Digital Mock-Up) using the new designed components. If conflicts are detected the involved components have to be improved by the design engineers. Otherwise, they are passed to the next step in the CoC process.

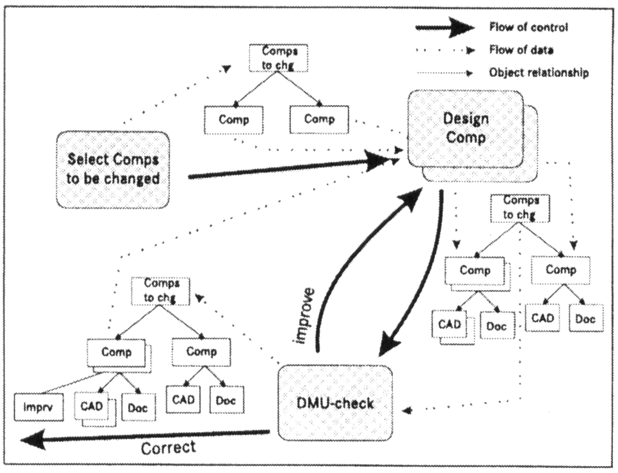

Figure 1: The change of components process

It is important to recognise that a WfMS supporting engineering processes can only be introduced successfully if it does not restrict the creativity of the design engineers. The WfMS must not enforce execution sequences of process steps which do not harmonise with the individual working behaviours of the design engineers involved. Therefore, one important feature of the **WEP** approach is *to support unstructured phases within a structured process.*

To support flexibility alone is not sufficient, however, to adequately serve this application domain (Beuter and Dadam 1996). To achieve reduction of development times also the traditional working behaviour of the design engineers has to be changed: (1) Final but also intermediate results should be made available (if reasonable) at earlier points in time. For example, for a DMU-check only the envelope of a changed component must be known, its internal structure is not necessary, it can be designed later. (2) More team-oriented cooperation among all persons involved must be achieved in order to detect errors and mismatching designs in earlier phases.

Aspect (1) allows to *simultaneously work* on different aspects of a design but also includes the risk to work with obsolete data. Therefore, mechanisms have to be provided which control the usage of preliminary data. Aspect (2) means that people (perhaps virtually) meet from time to time to discuss and synchronise their work. During such phases new commonly accepted design versions are determined which have to be made available to all persons involved. – The **WEP** approach supports both aspects.

All three aspects "no restriction of individual creativity", "release and control of preliminary results", and "support of coordination phases" are not provided by current WfMS technology (cf. e.g. Jablonski and Bussler 1996). The paper describes how WfMS technology can be augmented to provide adequate workflow management support for engineering processes.

The remainder of the paper is organised as follows: In section 2 the philosophy behind the **WEP** approach is outlined. It is also described how requirements elaborated above are handled. Section 3 discusses related work. The paper is finished by a summary and outlook.

2 THE WEP MODEL

The **WEP** approach augments the functionality of a process-oriented WfMS by additional features called *creativity support*, *simultaneous engineering support*, and *consolidation support*.

The creativity support is realised in the **WEP** model by (possibly) long-running *goal-oriented activities*; each of which representing a weak structured or unstructured subprocess. Like other process steps goal-oriented activities can be combined to a superior engineering process in the usual process-oriented way. Processes enclosed in a goal-oriented activity can be called in any order and as often as needed by the associated participant (called the "role filler" in WfMS terminology). Instead of modelling the sequence of processes explicitly, goals are specified which describe the task the participant has to perform. A goal is always associated with a point in time at which a result has to be delivered in a specified quality (**WEP** distinguishes different kinds of data quality reflecting the different phases of the design). Therefore, a goal is also called *milestone* in the **WEP** model. Data qualities are specified by logical expressions.

This goal-oriented description makes it possible that a participant of a goal-oriented activity can pass (preliminary) data to successor steps of the process before the task has been completed. As a consequence, participants assigned to sequential process steps can already use these provisional results to start their tasks. During these so called *Simultaneous Engineering phases* errors and mismatching designs can often be detected based on these preliminary data. The "data producer" receives the information earlier and time-intensive iterations of the process can be avoided. On the other hand, simultaneous engineering potentially jeopardises data consistency because different data versions can be created simultaneously. So mechanisms are provided by the **WEP** model in order to resolve data inconsistencies (see section 2.2.3).

2.1 Modelling Goal-oriented Activities

Goal-oriented activities are basic units in the **WEP** model. Similar to other process-oriented WfMS they are connected by control and data flow arrows (and described using a graphical editor, as usual). Legal activity connections are based on the specification of the activities' input and output objects. Due to the new features the **WEP** WfMS offers some additional properties have to be modelled.

2.1.1 Specification of Output Objects

As mentioned above, milestones are used to specify output objects. Output objects can be released in a provisional or in a final state. Among other things (see figure 2) the following properties have to be specified:

- *Quality Level:* It describes the output object's data quality. The quality level of an output object is based on the *quality levels* of the object's type. When a new object type is defined different quality levels are specified, too. They are based on logical expressions using the object's attributes and relationships. During the specification of an activity's output objects additional logic expressions can be used to specify finer nuances within an object's quality level. Due to space limitations no formal description of quality levels are presented here.

- *Time limit:* A *"point in time"* must be modelled by which the object in the data quality specified has to be made available to succeeding process steps. Each goal-oriented activity is associated with one "activity-global" *time reference point*, called *activity reference point* in the **WEP** model. All milestones *within* a goal-oriented activity are defined relative to the associated activity reference point. The activity reference points, in turn, are related to global deadlines which are valid for the workflow as a whole (*workflow reference points*).

```
OUT <ObjName> : <ObjClass>
  REQUIRED|OPTIONAL IN QUALITYLEVEL  <Name> :
             {TIMEREFPOINT  |  {+|-}<TimeSpec>}
  BASED ON OBJECT QUALITYLEVEL <QltyName>
  [boolExpr ... boolExpr]
  CONCURRENTMODE : CONSOLIDATION|AUTONOMOUS
  [DISTANCE : <Number>]
  ...
```

Figure 2: Specification of an activity's output objects (keywords in uppercase letters)

2.1.2 *Specification of Input Objects*

Due to the possibility of releasing preliminary results some more properties must be specified for the activity's input objects (see figure 3):

- *Quality range:* This property describes the minimal and maximal data quality needed for the activation of an activity.
- *consistency policy UNDO_REDO/MERGE:* This property specifies how new results (e.g. refined objects with higher data quality level) which have been passed to a running activity shall be integrated. Section 2.2.3 describes the strategies offered by the **WEP** model in more detail.

```
IN REQUIRED|OPTIONAL <ObjName> : <ObjClass>
   [QUALITYRANGE "]"|"["QltyLvl;QltyLvl"]"|"["]
   CONSISTENCYPOLICY UNDO_REDO | MERGE
...
```

Figure 3: Specification of an activity's input objects

Figure 4 gives an example of the input/output specification of the activities *Design Comp* and *DMU-check* (see figure 1).

```
ACTIVITY Design Comp
...
OUT comp : CompClass
  REQUIRED IN QUALITYLEVEL EnvelopeSpecified :
                                     - 3 WEEKS
    BASED ON OBJECT QUALITYLEVEL CompEnvelSpec
    CONCURRENTMODE : CONSOLIDATION
  ...
  REQUIRED IN QUALITYLEVEL DesignFinished :
                                     TIMEREFPOINT
    BASED ON OBJECT QUALITYLEVEL CompFinished
    // additional to object's quality level
    // CompFinished changes must be documented
    // i.e. relationship to Doc object exist
    comp.IsDocumentedBy == DEFINED
    CONCURRENTMODE : AUTONOMOUS

ACTIVITY DMU-Check
IN REQUIRED comp : CompClass
QUALITYRANGE [EnvelopeSpecified;DesignFinished]
CONSISTENCYPOLICY MERGE
...
```

Figure 4: Example for input/output specification

2.2 User Interactions

The additional features described above lead to some extra effort at build time. As we will see, they allow forms of interactions at runtime which are not supported by traditional WfMS.

2.2.1 *Overview*

Figure 5 shows a typical interaction sequence that is possible for users of the **WEP** runtime environment. Due to limitations of space only the most interesting operations are explicitly explained here.

A new workflow instance is initiated by the operation *InitActivity(ActivityID, Time, ...)*. Here logical times specified at build time are replaced by real times. "Watchdogs" watching the different milestones are started and process steps ready for activation are inserted into worklists of the affected participants.

As already mentioned above, a goal-oriented activity, in contrast to a basic step in a traditional WfMS, represents a more complex and unstructured task. Usually, a participant needs some days or weeks for processing a task and uses different tools during that time. The tools can be called in any order and as often as needed.

A user may work with different versions of an object at the same time, all of which derived from the same input object. Think of trying out different design alternatives. Only one of these versions will be pre-released or released to other participants. In the **WEP** approach this typical working style is supported by the *U-mask* metaphor (see figure 5) which provides a (read-only) data input area, a private work area, and a (public) output area for the user's objects. The user explicitly transfers objects from the work area to the output area. The system controls (and may refuse the transfer) whether the object satisfies the pre-modelled object quality. Once an object is successfully transferred into the output area, the **WEP** system checks the input conditions of succeeding process steps and offers them for activation, if possible. It also informs the affected users when newer input objects are available. – By doing so, the **WEP** system actively supports the important *simultaneous engineering phase (SE-phase)* in a controlled way.

2.2.2 *Simultaneous Engineering Phases*

A great advantage of the **WEP** concept is that sequentially modelled process steps can be processed time shared to a certain extend due to the possibility of an advanced data release. By these SE-phases the running time of an engineering process can be significantly decreased. During a SE-phase the **WEP** runtime environment is monitoring all activities relying on pre-released data objects. Similar to groupware systems (Ellis et al. 1991), it organises *consolidation phases* where the simultaneously working users can discuss and synchronise new object versions. It supports the integration of object versions created by simultaneously and autonomously performed activities (see section 2.2.3).

Figure 6 describes in detail a SE-phase based on the activities of figure 5.

A SE-phase is implicitly started once an activity (e.g. activity B in figure 6) pre-releases one or more data objects. An activity D using pre-released data is called "a dependent activity of B" (D depAct B in figure 6). Both activities create *simultaneously* new versions ($V_{01}^{Q_i}(B)$,

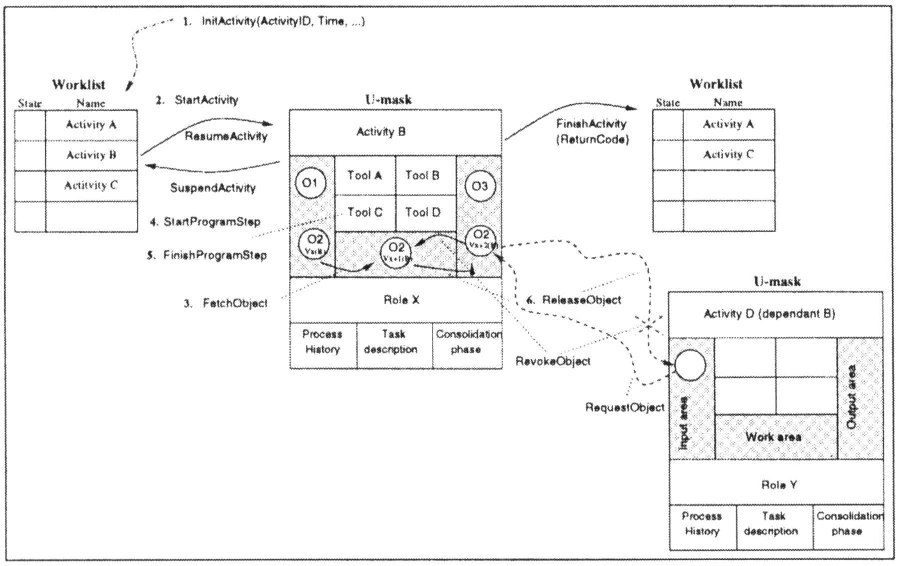

Figure 5: Interaction sequences in the **WEP** WfMS

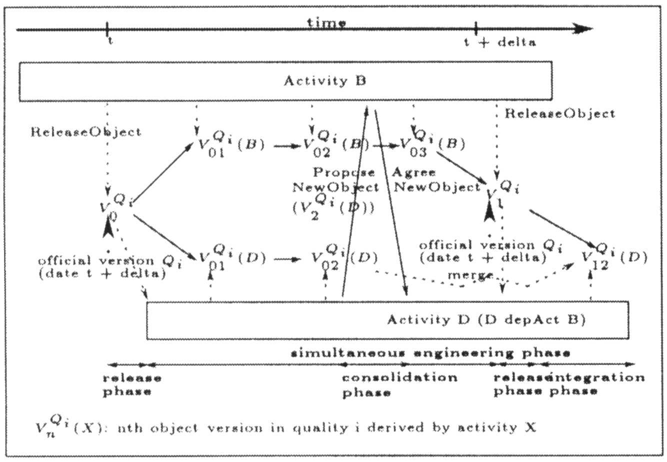

Figure 6: Simultaneous engineering phase

$V_{02}^{Q_i}(B)$, $V_{01}^{Q_i}(D)$, $V_{02}^{Q_i}(D)$) all derived from the object's *official version* ($V_0^{Q_i}$). Each involved participant can start a *consolidation phase* by calling the operation *ProposeNewObject*.

The **WEP** WfMS provides the basic mechanisms for a distributed consolidation phase: ProposeNewObject, FinishNegotiation, AgreeNewObject, RejectNewObject. The restriction to these elementary features makes the **WEP** model broadly applicable: depending on the system resources available, a consolidation phase may be implemented by simple email exchange, by additional cooperation operations (Mitschang et al. 1996), or by the integration of teleconferencing systems (Weber et al. 1997).

2.2.3 Integration Phases

SE-phases potentially jeopardise data consistency because different object versions can be created simultaneously. Therefore, a WfMS supporting SE-phases must also provide mechanisms to detect and resolve inconsistencies.

Firstly, it is important to inform all affected users about the availability of new object versions in preliminary or of final data quality. Secondly, the users must determine whether their work performed so far is affected by potential changes in these objects.

In many cases, the affected users have to integrate the new objects appropriately. In simple cases, e.g. where program-based computations are performed, a simple repetition of the task with the new data and "intellectual" (or even program-based) comparison of the result is sufficient. In most cases, however, an "intellectual" merge of versions has be performed by the affected user. Some progressive CAD tools, for example, support the visualisation of differences in object versions. So where necessary, the user can merge different object versions with little effort.

3 RELATED WORK

One great disadvantage of todays workflow management technology is the lack of flexibility at runtime restricting its applicability in several domains (e.g. Kuhn et al. 1994, Mitschang et al. 1996). Therefore, diverse WfM approaches are proposed offering more flexibility at runtime (Eder and Liebhart 1995, Reichert and Dadam 1998, Antunes et al. 1995, Weber et al. 1997). But none of the proposed concepts is really applicable to the automotive engineering domain: In (Eder and Liebhart 1995) flexibility is limited to the users' possibility to "recall" an already passed workflow instance. Additionally, (Reichert and Dadam 1998) allow to permute sequential process steps or to omit some steps at all. These approaches assume that there is one standard process which should normally be enforced by the WfMS. Deviating from this standard way should be an exception. Engineering processes contain subprocesses where no standard way exists. Hence, the **WEP** approach introduces goal-oriented activities where no flow of control is defined. In order to assist the user, a data-centred approach (Bredenfeld 1995) is used for, instead, defining the preliminary and final milestones of the activity. In contrast to other data centric approaches this style of modelling is only applied within a goal-oriented activity. Therefore, the usual disadvantage of data centric approaches (very complex, difficult to administer) can be avoided.

Other approaches include groupware functionality into their WfMS (e.g. Weber et al. 1997; Antunes et al. 1995) to enable users to enact at the same time while they are spatially dispersed. These groupware features can also be used for supporting unstructured work (called meeting in Weber et al. 1997) at specific points in a workflow. Since the internal structure of a meeting is usually not known by the WfMS the meeting's results can only rudimentarely be integrated into the flow of control. The basic mechanisms for such meetings is also offered by the **WEP** model to support consolidation phases (see section 2.2.2). So these approaches can be used in combination with our model.

None of the WfMS models known to the authors supports the release of preliminary data.

Design Flow Management Systems (DFMS) are also proposed for the process support in the engineering domain. DFMSs allow a hierarchical refinement and decomposition of a design task (Mitschang et al. 1996; Maurer 1996; Goldmann 1996). The engineering process (from a first idea to complete, proved, and released CAD Design models) is not directly supported (Maurer 1996; Goldmann 1996) or can only be modelled by conventional workflow features (Mitschang et al. 1996). At Daimler-Benz the decomposition is usually done in teams which consist of people involved in the engineering process. Different decomposition variants are discussed, and a CoC process (cf. figure 1) is triggered for a stepwise concretion of design tasks. From this point of view, a DFMS and the **WEP** model can be used complementarily.

4 SUMMARY AND OUTLOOK

The **WEP** model introduces the additional features, "integration of goal-oriented activities for the support of unstructured creative subprocesses", "release and control of preliminary data", and "support of coordination phases", which are not available in other approaches. Thus, it provides an adequate support for engineering processes both at workflow buildtime and at workflow runtime.

At present the concepts described above are not implemented yet. But, in the next future a prototype implementation of the **WEP** WfMS is planned in order to show its applicability. The implementation will be based on the commercially available product data management system Metaphase which provides the data version features needed. The **WEP** user interface will be implemented in Java.

ACKNOWLEDGEMENTS

I would like to thank Carmen Winter for many helpful suggestions concerning the style of this paper.

REFERENCES

Antunes, P., N. Guimaraes, J. Segovai, and J. Cardenosa (1995, August). Beyond formal processes: Augmenting workflow with group interaction techniques. In *Proc. Int'l Conf. on Organizational Computing Systems (COOCS '95*, San Jose, California.

Beuter, T. and P. Dadam (1996, June). Application specific requirements for workflow management systems. Technical Report 96-04, University of Ulm (in German).

Bredenfeld, A. (1995, September). Cooperative concurrency control for design environments. In *Proc. of the European Design Automation Conference*, Brighton.

Eder, J. and W. Liebhart (1995, May). The workflow activity model WAMO. In *Proc. 3rd Int. Conf. on Cooperative Information Systems*, Vienna, Austria, pp. 87–98.

Ellis, C. A., S. J. Gibbs, and G. L. Rein (1991, January). Groupware – some issues and experiences. *CACM 34*(1), 39–58.

Goldmann, S. (1996, June). Procura: A project management model of concurrent planning and design. See WET-ICE (1996), pp. 177–183.

Jablonski, S. and C. Bussler (1996). *Workflow Management - Modeling Concepts, Architecture and Implementation.* Thompson Computer Press.

Kuhn, K., M. Reichert, T. Beuter, M. Nathe, and P. Dadam (1994, November). An infrastructure for cooperation and communication in an advanced clinical information system. In *Proc. 18th Annual Symposium on Computer Applications in Medical Care (SCAMC'94)*, Washington, D.C., pp. 519–523.

Maurer, F. (1996, June). Project coordination in design processses. See WET-ICE (1996), pp. 191–197.

Mitschang, B., T. Hrder, and N. Ritter (1996, February). Design management in concord: Combining transaction management, workflow management and cooperation control. In *Proc. 6th Int'l Workshop on Research Issues in Data Engineering: Interoperability in Multidatabase Systems (RIDE-IMS)*, New Orleans.

Reichert, M. and P. Dadam (1998, March). $ADEPT_{flex}$ – supporting dynamic changes of workflows without loosing control. *Journal of Intelligent Information Systems (JIIS), Special Issue on Workflow and Process Management 10*(2).

Weber, M., G. Patsch, A. Scheller-Huoy, J. Schweitzer, and G. Schneider (1997, January). Flexible real-time meeting support for workflow management systems. In *Proc. 30th Hawaii In'l Conf. on System Sciences HICSS*, Maui, Hawaii.

WET-ICE (1996, June). *Proceedings of WET-ICE 1996*, Stanford, Ca, USA. IEEE.