WFESelector
A Tool for Comparing and Selecting Workflow Engines

Karim Baïna
ENSIAS, Université Mohammed V - Souissi, BP 713 Agdal - Rabat, Morocco

Keywords: Workflow management systems, Workflow management systems evaluation, Workflow management systems comparison criteria.

Abstract: The task of selecting a workflow engine becomes more and more complex and risky. For this reason, organisations require a broad, and a clear vision of which workflow engines are, and will continue to be, suitable for changing requirements. This paper presents a workflow engines comparison model to analyse, compare, and select business process management modelling and enactment engines (Workflow Engines or WFEs) according to user specific requirements. After the description of the underlying model itself, we present the implementation of this workflow engines comparison model through our multi-criteria workflow engines comparison and selection prototype WFESelector. The later proposes two scenarios for selecting relevant WFE : either to express dynamically multi-criteria query upon a WFE evaluation database, or to browse the whole WFE classification through a reporting aggregation based dashboard. WFESelector is subsequently experimented to assess criteria satisfaction on a very large number of open source workflow engines (as numerous as 35).

1 INTRODUCTION

Enterprise business models are becoming more and more complex, involving numerous interacting applications within rich business and technical contexts. Thus, enterprise business processes inherit this business models growing complexity. Since those business processes are considered in the enterprise business core strategy, they need workflow engines that handle suitably the continuously growing business processes management complexity. Nowadays, business process management engines and Workflow Management Engines (WFE) become numerous, heterogeneous, and provide several functionalities which make the task of selecting a WFE complex and risky. For this reason, organisations require a broad, and a clear vision of which workflow engines are, and will continue to be, suitable for changing requirements. Selection of such workflow engines is strategic for nowadays organisations.

Our contribution is at one side, to present our workflow engine comparison criteria model, and its implementation within WFESelector prototype, our multi-criteria workflow engines comparison and selection tool, and at the other side, to experiment the strength of our approach and prototype. The paper is organised as follows : section 2 discusses related works, section 3 presents our workflow engines comparison model, section 4 illustrates the approach through the results of our workflow engines comparison model experiment, and section 5 concludes and gives an outlook for this work.

2 RELATED WORKS

Many other works have studied the problem of producing a generic workflow engine comparison model, but none of them has proposed a rich workflow comparison model as we propose (35 comparison subcriteria). (Lei and Singh, 1997) compares workflow engines according to their process definition metamodels through eight criteria : Granularity, Control Flow, Data Flow, Organisational Model, Role

Baïna K. (2007).
WFESelector - A Tool for Comparing and Selecting Workflow Engines.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - DISI*, pages 330-337
DOI: 10.5220/0002397303300337

Binding, Exception Handling, Transaction Support, and Commitment Support. (M. Rosemann and zur Muehlen, 1998) compares workflow engines according to their organizational meta model, and Process Meta Model richness and expressivity. (Van Der Aalst et al., 2003) tackles the comparsion problem from the point of view of the ability of workflow engines to support workflow patterns. (Yu and Buyya, 2005) bases its comparison of scientific workflow engines on four criteria : Workflow Design, Workflow Scheduling, Fault Tolerance, and Data Movement. (Stoilova and Stoilov, 2006) focuses its comparison on a subset of five sub-criteria of (McCall et al., 1977) software quality criteria : Reliability, Usability, Efficiency, Maintainability, and Portability.

3 WORKFLOW ENGINES COMPARISON MODEL

Our workflow engines comparison model will be presented in three steps : first, section 3.1 present the chosen criteria for workflow engines comparison, then section 3.2 shows how key performance indicators are used to evaluate presented criteria, and finaly section 3.3 summarizes our approach through the presentation of a workflow engines comparison metamodel.

3.1 Workflow Engines Comparison Criteria Model

WFESelector bases its Workflow engine comparison upon 31 criteria and sub-criteria clustered in three sub-classes as shown in figure 1: (1) 12 *executability criteria*, (2) 10 *vision criteria*, and (3) 9 *contextual criteria*. Those criteria and their sub-criteria are derived from criteria proposed by WARIA (Workflow And Reengineering International Association) (war,), from criteria highlighted by workflow research (Reichert and Dadam, 1997; Bernstein et al., 1999; Russell et al., 2005; Gaaloul et al., 2005; Baïna et al., 2006; Gaaloul and Godart, 2005) from software quality factors defined by McCall et al. (McCall et al., 1977), and finally from auxiliary contextual properties on Workflow engines. These criteria are summarised in figure 1 and described below.

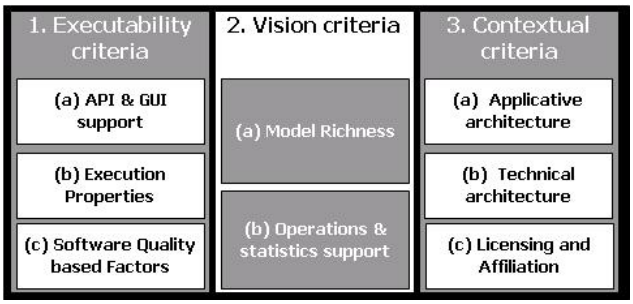

Figure 1: Workflow Engine Criteria Classification.

1. **Executability criteria:**
   (a) **API & GUI support:**
      i. **Activity Programming (***API Proposition - WfMC, OMG, Wf-XML, SWAP, etc.***-)** The API is the cornerstone of activity programming, with support for deadlines management, and events mechanisms: Facilities Provided by the Engine, API Completeness, and Events Processing Mechanisms ;
      ii. **Internet Support (***Web/Internet oriented GUI***)** The user interface can be operated from a simple Web browser. This covers the following functions: worklist handler, activity execution, graphical procedure status, and starting a procedure instance.
   (b) **Execution Properties:**
      i. **Throughput Rates (***Efficiency***)** Depends on the efficiency of the engine, and its capacity to run on top of a distributed cluster of servers: Server Scalability, Distributed architecture, and Client Implementation ;
      ii. **Distribution (***MOM support, asynchronous messages support, triggers/notifications) support* Supports co-operation, within the same company or set of partners, of several independent Workflow engines operated by different departments (e.g. a manufacturing process that triggers a purchase order process, and cascades to an Extranet based supply chain process): Import/Export, and Automatic Co-operation Mechanism ;
      iii. **Dynamic Changes** *(Modification of process model at runtime)* (Reichert and Dadam, 1997) Collaborative and ad-hoc procedures, even if known in advance, must be defined on the spot (*adaptive workflow* (Bernstein et al., 1999)). Since they might be incomplete, it is important to be able to change them dynamically to factor in unforeseen situations: Change Procedure Network Definition, Change Variables Definition, Change Dispatching Rules, Late Sub-Network Definition, and Change Activity Implementation ;

iv. **Procedure Definition** *(Maniability and usability)* The best way to shorten process definition time is to offer a graphical process definition tool. Further assistance can be provided if the Workflow engine supports additional features like skip, undo, offered/accepted states, suspend, delegate and reroute: Graphical Definition, Assisted Definition of Rules-/conditions, Embedded Feature, BPR and Simulation ;

v. **Ready-to-use agents (***Maturity and executability***)** The engine is provided together with ready-to-use agents. No programming is required to run defined procedures. This is mandatory for ad-hoc Workflow applications and includes: a worklist handler, and a standard activity agent for managing the dialogue with the user: Worklist Handler, Procedure Network Visualizer, and Standard Activity Agent.

(c) **Software Quality based Factors:**

i. **Adaptability** *(Development framework proposition)* The engine minimises the necessary modification effort at requirement evolution ;

ii. **Maintainability** *(Code readability and documentation)* The engine minimises the effort to localise and correct bugs ;

iii. **Reusability** *(Extensibility)* The engine can be partially or totally used with an other application ;

iv. **Reliability** *(Exception management and fault tolerance)* The engine does what it is supposed to do, and what the user expects it to do, and it does so without breaking anything in the process: accuracy, error tolerance, consistency and simplicity in design. (Russell et al., 2005) details workflow execption as 5 types : work item failure, work item deadline expiry, resource unavailable, external trigger, and constraint violation).

v. **Testability** The engine facilitates test procedures.

2. **Vision criteria:**

(a) **Model Richness:**

i. **Procedure Power (***Richness and process model expressivness)* The capacity of the procedure development environment to express the real complexity of procedures in all their smallest details: Network Structure, Variables Definition, Exception Processing, Complexity Management, and Verification ;

ii. **Standards compliance** WfMC, OMG, IETF, (vision: BPMI, SOA, WARIA);

iii. **Data flow** variable containers I/O mapping, through a database/MOM/Broker/XML Bus, etc ;

iv. **Control flow** sequence, split (and/or/xor), join (and/or/xor), iteration, sub-Workflow, multiple activity occurrences, etc ;

v. **Transactional Flow** After a failure, transactional external transitions are fired by external entities (scheduler, human intervention, etc.) and allow to the failed activity to interact with the outside environment to recover a consistent state. The goal is to bring the failed process back to some semantically acceptable state. Thus, failure inconsistent state can then be fixed and the execution resumed with the hope that it will then complete successfully ;

vi. **Dispatching & Organization Representation** *(Advanced group management)* The capability to dispatch each individual activity to the participant with the correct level of access rights and appropriate role in the organization: Dispatching Rules, Organizational model, Administration and Privacy, Substitution Rules, and Import from Directories ;

vii. **Enterprise Application Integration (EAI) (***Interoperability***)** Evaluates the tools provided to take benefits of most modern Enterprise Application Integration: message queuing, application adapters, COM/DCOM distributed integration, CORBA and/or Web Services support, and 2PC transaction support ;

viii. **Activity Definition** *(Model driven or code driven)* Activity implementation (no programming) can be achieved by providing embedded support, libraries of activities, and forms generation tools integrated with the process definition tool. Additional flexibility can be provided by a library of basic actions that can be scripted (possibly graphically) to implement activities: Activity Library, Form Generation, Action Library, Action Scripting Assembly, Integration Tools, and Multi-lingual Support.

(b) **Operations & statistics support:**

i. **Workflow Logs (***Event log proposition***)** Evaluates the existence of WfMS logs storing events occurring during process execution. Activities are traceable, meaning that the sys-

tem should in one way or another keep track of ongoing and past executions ;

ii. **Reporting monitor proposition** All features that facilitate operation of the server, and collection and interpretation of statistics on case processing, including tools provided to minimize the cost of communications and workstation administration: Recovery and Restart, Statistics Recording, Operation Archiving, Statistics Processing, and Home Work.

3. **Contextual Criteria:**

(a) **Applicative architecture** The engine is well adapted to applications like e-government, e-commerce, SCM (Supply Chain Management), CRM (Custumer Relation Management), KM (Knowledge Management), QM (Quality Management), GIS (Geographical Information System), etc ;

(b) **Technical architecture:**

i. **Adopted standard** WfMC, OMG, IETF, (vision: BPMI, SOA, WARIA) ;

ii. **Adopted model** Workflow patterns[1] (Van Der Aalst et al., 2003), Petri Nets, Graph theory, process algebra, PERT, Gantt, BPEL, etc ;

iii. **Transactional model** During the Workflow execution, an activity can pass through several stages defined as activity states (aborted, failed and completed). The transactional properties of an activity depend on the set of its internal states transitions ;

iv. **Event log model** Richness of the log (in terms of information it contains), the log follows a standard format such as WfMC Audit Trail, etc.

(c) **Licensing and Affiliation:**

i. **Incubator/Initiative** OBJECT WEB, APACHE, etc ;

ii. **License** APACHE, MIT, IBM, LGPL, GPL, etc.

iii. **Relations and Networking (***Historical relations with other workflow engines***)** Links with other open source or commercial workflow engines ;

iv. **Web Site accessibility and richness** absent/not reliable/not referenced, up to date, reliable/rich, etc.

[1]Workflow patterns : www.workflowpatterns.com

3.2 Workflow Engines Comparison Key Performance Indicators Model

We can divide the presented criteria in two categories: those one may evaluate through a mark and those which are contextual and less subject to an effective mark. For example, what should mean a mark affected to *Licencing and Affiliation*? We consider that contextual criteria have not to be marked while execution properties and vision criteria have to be. Each high level criteria is seen as a hierarchical marked criterion that aggregates the set of all its submarked criterion marks trhough a criterion formula (in the same vain of multidimensional database *roll-up* aggregation operation). In the following, aggregation formulas of each hierarchical marked criterion given at section 3.1 will be presented and explained.

The criterion mark of a hierarchical marked criterion is computed as the uniform average of its sub marked criterion (which means that each atomic sub marked criterion has the same weigth - e.g. each *Executability* marked sub criterion has the same weigth $\frac{1}{12}$ with 12 the number of *Executability* marked sub criteria, and each *Vision* marked sub criterion has the same weigth $\frac{1}{10}$ with 10 the number of *Vision* marked sub criteria). For formulas simplicity, each marked criterion name will represent, by language abuse, its evaluation given floating mark (e.g. $Executability(w \in WFE)$ denotes Executability criterion mark of the WFE o. For simple visualisation tuning, h_s will denote a 2D (Executabilty $\times$ Vision) space homothety transformation scale, and $(-t_e, -t_v)$ will denote a 2D space origin translation.

1. Executability criterion mark is the uniform average of 2-sub-criteria of API & GUI support, 5-sub-criteria of Execution Properties, and 5-sub-criteria of Software Quality.

$$Executability(w \in WFE) = \left(\left(\frac{2 * API\ \&\ GUI\ support + 5 * Execution\ Properties + 5 * Software\ Quality}{12}\right)(w) * h_s\right) - t_e$$

(a) $$API\ \&\ GUI\ support(w \in WFE) = \left(\frac{Activity\ Programming + Internet\ Support}{2}\right)(w)$$

(b) $$Execution\ Properties(w \in WFE) = \left(\frac{Throughput\ Rates + Distribution + Dynamic\ Changes + Procedure\ Definition}{5}\right) + \left(\frac{Ready\text{-}to\text{-}use\ agents}{5}\right)(w)$$

(c) $$Software\ Quality(w \in WFE) = \left(\frac{Adaptability + Maintainability + Reusability + Reliability + Testability}{5}\right)(w)$$

2. Vision criterion mark is the uniform average of 8-sub-criteria of Model Richness, and 2-sub-criteria of Operations & statistics support.

$$Vision(w \in WFE) = \left(\left(\frac{8 * Model\ Richness + 2 * Operations\ \&\ statistics\ support}{10}\right)(w) * h_s\right) - t_v$$

(a) $$Model\ Richness(w \in WFE) = \left(\frac{Procedure\ Power + Standards\ compliance + Data\ flow + Control\ flow}{8}\right)$$

$$+\left(\frac{Transactional\ Flow + Dispatching\ \&\ Organization\ Representation}{8}\right)$$
$$+\left(\frac{Enterprise\ Application\ Integration + Activity\ Definition}{8}\right)(w)$$

(b) $Operations\ \&\ statistics\ support(w \in WFE) =$
$\left(\frac{Workflow\ Logs + Reporting\ monitor\ proposition}{2}\right)(w)$

3.3 Workflow Engines Comparison Metamodel

Now that we have presented the criteria and their evaluation we can present our workflow engines comparison metamodel which is composed of 6 meta-concepts, as shown in figure 2):

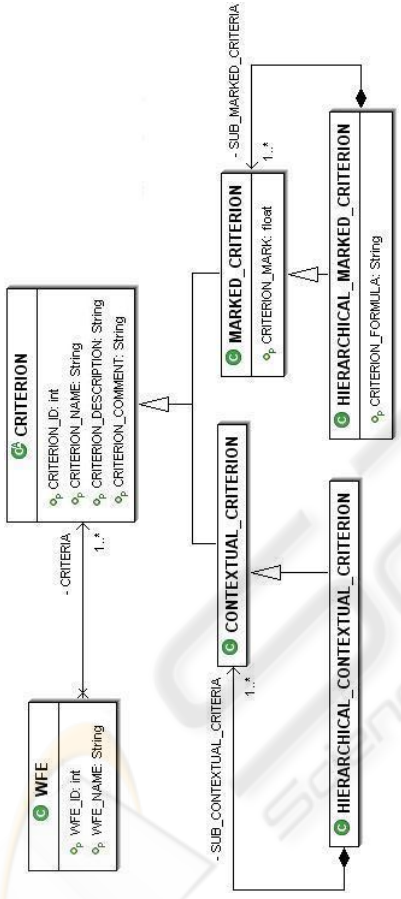

Figure 2: WFESelector reporting Metamodel.

- WFE meta-concept: represents a workflow engine entity with its WFE_ID, and WFE_NAME ;
- CRITERION meta-concept: represents an abstract workflow engine criterion entity with its CRITERION_ID, CRITERION_NAME, CRITERION_DESCRIPTION, and given evaluation CRITERION_COMMENT;
- CONTEXTUAL_CRITERION meta-concept: represents a concrete workflow engine textual criterion entity that describes the evaluation result of WFE according to a contextual property ;
- HIERARCHICAL_CONTEXTUAL_CRITERION meta-concept: represents a hierarchical concrete workflow engine contextual criterion entity that is composed of many CONTEXTUAL_CRITERION entities ;
- MARKED_CRITERION meta-concept: represents a concrete workflow engine criterion entity with its given evaluation floating CRITERION_MARK ;
- HIERARCHICAL_MARKED_CRITERION meta-concept: represents a hierarchical concrete workflow engine marked criterion entity that is composed of many MARKED_CRITERION entities. This hierarchy aggregates the set of all its SUB_MARKED_CRITERION.CRITERION_MARK through CRITERION_FORMULA (in the same vain of multidimensional database *roll-up* aggregation operation).

4 WORKFLOW ENGINES COMPARISON MODEL EXPERIMENT

4.1 Workflow Engines Comparison Model Implementation: WFESelector

Figure 3 presents the applicative architecture of WFESelector of four layers.

Figure 3: Applicative architecture of WFESelector.

The **evaluation database layer** manages a relational database that stores for each studied workflow engine its evaluation information (i.e. the set of contextual and marked criteria). Those WFE evaluation marks are based upon studying their related research

papers, marketing white papers, and slides, setting up and testing technically all engines according to a complete case study process (in our case, the case study was based on ISO 9002 preventive, and corrective actions circulation process). Evaluation of a WFE is achieved by giving subjective marks: for each WFE, and CRITERION a floating evaluation MARK is given from 1 to 5 (1 is the worst mark, and 5 is the best mark). The **Data Mart layer** manages a multidimentional hypercube that stores for each studied workflow engine its aggregated marks and key performance indicators on the basis of the evaluation database information. The **Querying engine layer** provides an API to query relevantly either the evaluation database and the data mart. The **Reporting Dashboard layer** proposes synthetic visual reports on workflow engines key performane indicators.

WFESelector user, wanting to select a WFE according to specific requirements, has, in fact, two possible scenarios : (S1) to select some WFE among those present in WFESelector evaluation database in order to classify them function of their marked critera and sub-criteria (through a reporting dashboard), or (S2) directely to express a multi-criteria query based on the hierarchical criteria model and to obtain a list of classified WFE (through a querying engine).

As a first validation step, we have experimented our WFESelector prototype on a set of open source workflow engines. Nowadays, WFESelector gathers a database of 35 WFE (experimentation sample). Without loosing in generality, we present an execution experiment of WFESelector execution scenario (S1) in which the user selects 9 significant WFE (analysis sample) among WFESelector evaluated WFE. The following figures 4 and 5, and table classify graphically, and describe textually those queried 9 WFE in detail.

WFESelector scenario execution experiment results informs that generally, priorities taken into account by open source workflow engines software editors, are nowadays: (1) API & GUI support is the most common functionality; then (2) the model richness, then (3) the execution properties; then (4) software quality factors; and finally (5) Operations and statitics.

Moreover, WFESelector scenario execution experiment shows that Executability KPI is not generally dependent of Vision KPI. Four WFE classes are distinguished: (1) class of WFE whose Executability and Vision are correlated (SHARK and jBPM); (2) class of WFE whose Executability is more important than Vision (Jfolder and OPEN WFE); WFE whose Vision is more important than Executability (Bigbross Bossa, Lenya, and Runa); and (4) (OSWORKFLOW and XFlow) seams to be an isolated class from the the other 8 WFE. The user can then conclude from his/her first WFESelector scenario use that SHARK and jBPM are in the top level distinction, and that OSWORKFLOW and XFlow are apparently atypical within the 9 WFE that the user selected as analysis sample.

WFE	VISION	(2.a) Model Richness	(2.b) Operations & statistics
BIGBROSS BOSSA	7	3,25	2
JBPM	9	3,5	3
OPEN WFE	7	2,75	3,5
RUNA	8	3,375	2
SHARK	8	3,25	2,5

WFE	EXECUTABILITY	(1.a) API & GUI support	(1.b) Execution properties	(1.c) Software quality
BIGBROSS BOSSA	5	4	2,6	2
JBPM	9	4	3,4	3,2
OPEN WFE	8	4,5	2,6	3,2
RUNA	6	4	2,8	2,4
SHARK	8	4,5	2,8	3

Figure 4: WFESelector Pivot Table, and 2D results examples of reporting dashboard GUI.

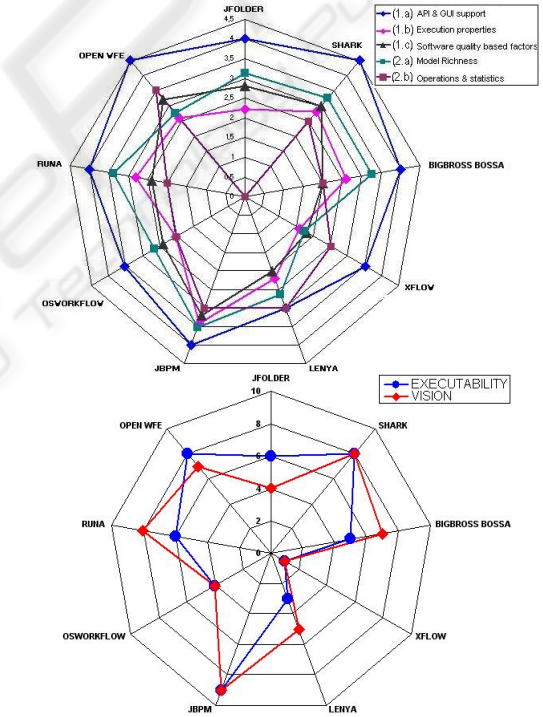

Figure 5: WFESelector Radar Graph results examples of reporting dashboard GUI.

5 CONCLUSION

This paper has presented our workflow engines comparison model for analysing, classifying, comparing, and selecting workflow engines according to specific requirements. The model has been implemented within WFESelector prototype. Our workflow engines comparison model has been experimented upon a set of 35 open source workflow engines, and has revealed preliminary criteria based classifications among open source workflow engines.

Our ongoing work involves the integration of a thin time dimension into the WFESelector *Data Mart* in order to track the evolution of WFE. Another actual outlook is related to the integration of Data mining within WFESelector *reporting process*. For example, WFESelector should help in automatic selection of evaluation weigths for roll-up operation instead of taking uniform average. We also aim to integrate WFESelector within a broader Workflow engines portfolio management platform.

In summary we believe that, once the research and development work on the aspects described above has been completed, this approach will result in a comprehensive platform that can substantially reduce (i) WFE presenting and understanding effort both for editors and customers, and (ii) WFE selection effort and therefore foster the widespread adoption of either open source or commercial workflow technology.

ACKNOWLEDGEMENTS

The author thanks all 2006 and 2007 ENSIAS engineers that participated with their experimental developments, and evaluations in WFESelector Project.

REFERENCES

Apache Lenya. http://lenya.apache.org.

Bigbross Bossa. http://www.bigbross.com/bossa.

Enhydra Shark. http://www.enhydra.org/workflow/shark.

jBPM. http://www.jbpm.org.

JFolder. http://www.jfolder.com.

OPEN WFE. http://web.openwfe.org/display/openwfe/Home.

OpenSymphony OSWorkflow. http://www.opensymphony.com/osworkflow.

Runa. http://wf.runa.ru.

Waria workflow comparative study. http://www.waria.com/books/study-2004.htm.

XFlow. http://xflow.sourceforge.net.

Baïna, K., Benali, K., and Godart, C. (2006). DISCOBOLE: A service Architecture for Interconnecting Workflow Processes. *Computers in Industry - Special issue on Collaborative Environments for Concurrent Engineering, Elsevier Science Publisher*, pages 768–777.

Bernstein, A., Dellarocas, C., and Klein, M. (1999). Towards adaptive workflow systems: Cscw-98 workshop report. *SIGGROUP Bull.*, 20(2):54–56.

Gaaloul, W., Baïna, K., and Godart, C. (2005). Towards Mining Structural Workflow Patterns. In Andersen, K. V., Debenham, J. K., and Wagner, R., editors, *16th International Conference on Database and Expert Systems Applications (DEXA'05)*, volume 3588, pages 24–33, Copenhagen, Denmark. Springer-Verlag.

Gaaloul, W. and Godart, C. (2005). Mining workflow recovery from event based logs. In van der Aalst, W. M. P., Benatallah, B., Casati, F., and Curbera, F., editors, *Business Process Management*, volume 3649, pages 169–185.

Lei, K. and Singh, M. (1997). A comparison of workflow metamodels.

M. Rosemann, M. and zur Muehlen, M. (1998). Evaluation of Workflow Management Systems - a Meta Model Approach. *Australian Journal of Information Systems*, 6(1):103–116.

McCall, J., Richards, P., and Walters, G. (1977). Factors of software quality. *NTIS*, 3.

Reichert, M. and Dadam, P. (1997). A Framework for Dynamic Changes in Workflow Management Systems. In Wagner, R., editor, *8th International Workshop on Database and Expert Systems Applications (DEXA'97)*, pages 42–48, Toulouse, France. IEEE Computer Society Press.

Russell, N., van der Aalst, W. M. P., ter Hofstede, A. H. M., and Edmond, D. (2005). Workflow resource patterns: Identification, representation and tool support. In *CAiSE*, pages 216–232.

Stoilova, K. and Stoilov, T. (2006). Comparison of workflow software products. In *International Conference on Computer Systems and Technologies, CompSysTech'2006*.

Van Der Aalst, W. M. P., Ter Hofstede, A. H. M., Kiepuszewski, B., and Barros, A. P. (2003). Workflow patterns. *Distrib. Parallel Databases*, 14(1):5–51.

Yu, J. and Buyya, R. (2005). A taxonomy of scientific workflow systems for grid computing. *SIGMOD Record*, 34(3):44–49.

WFE_NAME	License	Pros & Cons
Bigbross Bossa (Big,)	GPL	Light weigth easy to use Petri Nets based WFE, manages user roles over activities. Log file contains information about persistent progression status that is reloaded at each transition
Enhydra Shark (Enh,)	(Object Web) LGPL	has an XPDL based process definition language, has a Wf-XML based process interface, is a model driven engine, is a mature modelling, enactement and administration engine, has a reliable role management. Possesses good process instance tracking, and visualisation. Has many graphical modelers (e.g. Jawe, Together), wrappers (e.g. Swish), is integrated by OFBiz, and can be deployed within Pentaho. Has No transactional properties handled in the process model
jBPM (JBP,)	(Object Web) LGPL	Eclipse plugin, has a rich API, enables process application extensibily Model driven engine, describes rigorously graphically business processes and origanisation roles (Graph Oriented Programming). Enables process definition to be modified at runtime, with logging process definiton versions
Jfolder (Jfo,)	LPDL	has a rich API, Web Base engine, JBOSS based, has no procedure programming environment, is a true model driven engine. No event logs, no reporting tools
Lenya (Apa,)	(Apache)Apache Software License	is a CMS specific WFE, is XML based, integrates Cocoon framework, and is supported by Apache Foundation with an Apache License. Lenya dataflow enables XML attachment to be created, modified, and removed. Lenya is web oriented, attached documents status can be verified, at each instant. Lenya users can be notified by e-mail for an eventual document validation. Lenya has a scheduler providing a log of achieved tasks on documents with their relative begining and ending dates
OPEN WFE (OPE,)	BSD	OMG Compliant with a majority of Wf patterns (Van Der Aalst et al., 2003). stores its work data either in XML files or in relational databases. Workflow Participant roles are handled as accounts within LDAP repository. Manages suitably workitems dispatching between participants
OSWorkflow (OSW,)	OpenSymphony Software License	OMG compliant, State Machine based Model, several interface mechanisms (SOAP, EJB, Java classes). Events Journal can be extracted, but no operation can be processed over those events. The Web version is somehow complex to interact with. Automatic Process centered, and coded driven process definition and management
Runa (Run,)		is cross-platform end user workflow environment for JBPM, is web oriented, has an interesting user group (swimlane) based task assignment
XFlow (XFl,)	(Apache)Apache Software License	based on JBoss, may be used through a Web Service directely via JMS, integrates Log4j. Is extensible and eases application integration. Provides flexibility in: designing process workflow, reconfiguring processes, and defining routing rules. Provides visibility into bottlenecks in a process workflow. Eases gathering statistics on the performance of participants in a process workflow. Has no procedure programming environnement, and is mainly coded driven