Evaluation of Workflow Modeling Solutions

Ewoud van der Vliet
ewoudv@ndervliet.nl

ABSTRACT
In this paper a framework for the evaluation of workflow modeling solutions is derived from workflow modeling concepts found in literature. The focus is on concepts useful in business process reengineering and application development. Six workflow modeling tools are evaluated using the framework.

Keywords
Workflow, Model, BPM, BPR, Modeling Tool

1. INTRODUCTION
This paper analyses workflow modeling methods and tools. This is done against two distinct backgrounds, Business Process Reengineering and Application Development, in which these process models are used:

The business process reengineering (BPR) background where workflow modeling is used to optimize processes within an organization. To do this, collaboration between process actors and process modelers is needed in order to correctly define and redefine processes within an organization. In this role, workflow diagrams and specifications must be both easy to create and easy to interpret. They should include all possible eventualities, although, to improve clarity, sometimes things are left out, this can be done safely, as long as the resulting model is specific enough for the purpose of the model. Workflows are often displayed in a manner which clearly shows what parts of the organization are involved. [SN00]

In Application development, workflow models are used as requirements specifications. In emerging technologies like service oriented architecture (SOA), workflows and the layout of the business processes they represent are becoming increasingly important. For use in information systems development workflow models have to be truly complete and everything (input, output, and precondition) has to be correctly and completely specified, else the resulting software will fail acceptance tests and correcting mistakes can be costly.

Because application development has shifted from data-aware to process aware in the last decade, IT applications are no longer objects that need to be organized around, they are part of the (dynamic) business process[Aa04].

The aim of this paper is to identify key issues which have to be kept in mind when selecting a method/tool to model workflows for use in either BPR, application development or both. From the business side, workflows are being modeled using “flow” diagrams, from the application development side; UML provides activity diagrams which can be used to model workflows. Petri-Nets are usually used by the scientific community to explore workflow configurations.

1.1 Research Setup
The paper starts by explaining the concept of workflow. Then the most important concepts affecting the choice for certain workflow modeling methods which have been found in peer reviewed literature are presented.

Articles which this paper refers to have been found by searching the following databases: Ingenta (www.ingenta.com), EBSCO’s Business Source Premier (www.ebsco.com), ACM digital library (www.acm.org) and Google Scholar (scholar.google.com). These databases were searched using search terms like: ‘Workflow’, ‘BPM’, ‘Business Process Modeling’, ‘BPR’ and ‘Workflow Modeling’. Google Scholar was mostly used to quickly find useful links to articles found in other databases. The concepts found in the various articles are grouped by their main objective which they aim to help accomplish most: Business process reengineering or Application Development.

From these issues, a framework is derived, which is then applied to 6 workflow modeling tools geared towards various tasks within the workflow spectrum (e.g. web services, application development, process analysis). This paper should serve as an eye-opener to people considering the use of workflow/business process models.

2. WORKFLOW
The Workflow Management Coalition (WfMC) [Aa04] defines workflow as: “The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.” Workflow modeling is most popular for office processes which involve documents being passed around. For this purpose Workflow Management Systems have been developed by multiple vendors. These can be fitted with a business process workflow, which supports and monitors the execution of the business process. Business Process Execution Language (BPEL) is a Workflow management System orchestration language, based on separate XML based services chained together to execute a workflow specified in BPEL [Mic05].

2.1 Workflow Model
Workflow models appear in various detail levels and forms. They are always process oriented. The difference between a business process model and a workflow model is that a workflow model models the user to application domain and lists all possible process outcomes relevant for the application, a business process model can involve interactions between humans and all possible outcomes do not have to be modeled in detail. A Business process model can also model human to
human relations and is usually less detailed than a workflow model. Petri-nets are widely seen as the scientific fully functional workflow modeling method. They are state based and colored Petri-nets can represent just about any domain [Aalst04]. Because of the difference between various tools and implementations, when looking at workflow modeling capabilities, it is not possible to separate languages from tools, because many popular tools implement their own language or fail to fully implement the language they claim to support. [RHA+06a] dive deeper into the subject of language and tool-language syntactic capabilities.

Other scientists have adapted UML activity diagrams to model workflows. [Esh02] The recently released UML 2.0 specification includes numerous improvements gearing UML activity diagrams more towards workflow modeling [RAH+06b] [Stö04]. However, most BPM toolsets have adapted other principles for workflow modeling, usually used in concert with UML software design diagrams.

There are over 350 commercially available tools [HR00] which use various UML, Flowchart and other Workflow techniques to model business processes and workflows which support them.

3. BUSINESS PROCESS REENGINEERING

Business process consultants usually aim at defining the (simplified) workflow using model techniques which can be easily understood, especially when the process is being mapped during a workshop session, where usually whiteboards and post-it notes are used to model a process. Process consultants argue that tools that set boundaries and procedures slow them down and limit their ability to perform process optimization in a meaningful way. Process consultants usually prefer tools which give them freedom and ease of use, like Post-it notes on a wall. They identify more with the socio-technical factor in process design [Mum94]. Grover and Malhotra [GM97] also realize that classic BPR (throwing the current process out and introducing a new one) is often not successful. They theorize that the change process and broad organizational involvement is much more important. The semantics and syntax of the modeling methods are usually shown to participants right before they start modeling, so the emphasis is on easy to learn methods.

Hommes and Reijswoud [homrei00] developed a framework to evaluate business process modeling techniques. They identify 3 quality properties which relate to modeling language properties and 4 properties which relate to the overall modeling technique:

Expressiveness

Capability for modeling different domains

Arbitrariness

Degree to which it is possible to model the same domain in different ways

Suitability

A method and its tools can be more or less suitable for a specific application domain.

Comprehensibility

The ease with which all stakeholders involved can interpret and use the modeling technique.

Coherence

The degree to which various sub-models integrate to form a whole.

Efficiency

The amount of resources (time, people) claimed

Effectiveness

How well does the modeling method achieve its goal?

These properties are still a bit vague. Becker et al [BRU00] attempt to define a more specific set of rules for process modeling languages. Their guideline of modeling framework (GoM) consists of 6 general guidelines:

Correctness

Does the model accurately (enough for the models purpose) represent the real world situation.

Relevance

The modeled processes must be defined well enough to represent the real-world situation well enough for the model’s purpose.

Economic Efficiency

The modeling procedure should not consume more resources than the improvements return.

Clarity

The model should be easy to interpret and should not be overwhelming in details.

Comparability

The model should be comparable to other modeling techniques so its knowledge can be used for other models as well.

Systematic Design

The modeling method should support a systematic buildup of models and meta-models.

These factors contribute to the model quality; their impact depends on the perspective in which the models are being used. For instance, a BPR model can have more emphasis on Clarity and relevance, while a model meant for application design needs more emphasis on correctness. Large, complex processes, for instance in large service companies benefit from increased systematic design.

3.1.1 Cognitive Effectiveness of Diagrams

If the cognitive effectiveness of workflow models increases, they will be more effective in communicating their message and therefore people can be involved in modeling more easily.

Cognitive Effectiveness of Diagrams depends on two things according to Moody [Moo06]:

Expressiveness

Capability for modeling different domains

Arbitrariness

Degree to which it is possible to model the same domain in different ways

Suitability

A method and its tools can be more or less suitable for a specific application domain.

Comprehensibility

The ease with which all stakeholders involved can interpret and use the modeling technique.

Coherence

The degree to which various sub-models integrate to form a whole.

Efficiency

The amount of resources (time, people) claimed

Effectiveness

How well does the modeling method achieve its goal?

These properties are still a bit vague. Becker et al [BRU00] attempt to define a more specific set of rules for process modeling languages. Their guideline of modeling framework (GoM) consists of 6 general guidelines:

Correctness

Does the model accurately (enough for the models purpose) represent the real world situation.

Relevance

The modeled processes must be defined well enough to represent the real-world situation well enough for the model’s purpose.

Economic Efficiency

The modeling procedure should not consume more resources than the improvements return.

Clarity

The model should be easy to interpret and should not be overwhelming in details.

Comparability

The model should be comparable to other modeling techniques so its knowledge can be used for other models as well.

Systematic Design

The modeling method should support a systematic buildup of models and meta-models.

These factors contribute to the model quality; their impact depends on the perspective in which the models are being used. For instance, a BPR model can have more emphasis on Clarity and relevance, while a model meant for application design needs more emphasis on correctness. Large, complex processes, for instance in large service companies benefit from increased systematic design.

3.1.1 Cognitive Effectiveness of Diagrams

If the cognitive effectiveness of workflow models increases, they will be more effective in communicating their message and therefore people can be involved in modeling more easily.

Cognitive Effectiveness of Diagrams depends on two things according to Moody [Moo06]:

Expressiveness

Capability for modeling different domains

Arbitrariness

Degree to which it is possible to model the same domain in different ways

Suitability

A method and its tools can be more or less suitable for a specific application domain.

Comprehensibility

The ease with which all stakeholders involved can interpret and use the modeling technique.

Coherence

The degree to which various sub-models integrate to form a whole.

Efficiency

The amount of resources (time, people) claimed

Effectiveness

How well does the modeling method achieve its goal?

These properties are still a bit vague. Becker et al [BRU00] attempt to define a more specific set of rules for process modeling languages. Their guideline of modeling framework (GoM) consists of 6 general guidelines:

Correctness

Does the model accurately (enough for the models purpose) represent the real world situation.

Relevance

The modeled processes must be defined well enough to represent the real-world situation well enough for the model’s purpose.

Economic Efficiency

The modeling procedure should not consume more resources than the improvements return.

Clarity

The model should be easy to interpret and should not be overwhelming in details.

Comparability

The model should be comparable to other modeling techniques so its knowledge can be used for other models as well.

Systematic Design

The modeling method should support a systematic buildup of models and meta-models.

These factors contribute to the model quality; their impact depends on the perspective in which the models are being used. For instance, a BPR model can have more emphasis on Clarity and relevance, while a model meant for application design needs more emphasis on correctness. Large, complex processes, for instance in large service companies benefit from increased systematic design.
The techniques a model employs to store information, and how humans process the combined graphical information.

For the first point he lists the methods and techniques available to represent information in graphics:
- Shape
- Color
- Size
- Value
- Orientation
- Texture

When combined with theories on Human Graphical Information Processing, especially on perception, Moody identifies principles which affect cognitive processing of diagrams [Moo06]:
- Discriminability – ability to identify diagram elements and discriminate them from the background and from each other.
- Modularity – dealing with complexity by logically splitting up the modeled domain into sub-parts.
- Emphasis – providing a way to indicate how important certain parts of the model are.
- Cognitive integration – providing a logical overview of how model elements and models relate to each other and enable the reader of a model domain to form a coherent image of the model domain.
- Perceptual Directness – how well do the model elements relate to common principles so the reader can easily relate model elements to common concepts
- Structure – How well are model elements which share a common property grouped together?
- Identification – How clearly does the model show its relation to the real-world domain represented by it?
- Visual Expressiveness – How many of the techniques available to represent information in graphics are used? Are multiple techniques used for the same information in order to speed up the cognitive process?
- Graphic Simplicity – How many conventions does the method have concerning the number of unique graphical elements?

3.1.2 Simulation
According to Becker et al. [BRU00] and Giaglis [Gia01], simulation is typically suited for BPR, this is an extra challenge for BPR, since simulation of course requires a complete and verified workflow model. Many BPR efforts use business process models, few of these models normally make it into complete workflows. However, modeling a current business process well enough to identify problems is an important part of the BPR process. Using smart algorithms, processes improvement suggestions can be automatically generated using the process model and simulation data [Gia01].

4. APPLICATION DEVELOPMENT
Application Developers use workflows to define software requirements. They value correct and detailed specification of workflows. They do not need to have information on organizational functions present in the model. Organizational (BPR) workflow models can be used as a starting point, but the tasks and input/output data is likely to be modified extensively before software solutions can be created using the documented workflow. The rise of service oriented architecture (SOA) has steered applications towards a workflow oriented architecture, with individual software services chained together to form a workflow. This makes it easy to integrate IT into organizational processes, as organizational models need little or no change to incorporate IT services. The perspective is similar. Other IT application architectures require the creation of separate models to model data-flow in applications. [BRU00]

4.1.1 Pattern Support
Russel et al. [RHA+06] have defined a number of process patterns, supported by colored Petri-nets, which were then evaluated in various workflow specification languages, including the input languages of various workflow management systems. Especially when used in IT systems, it is important that workflow models can accurately represent the reality of the business process. Many of the 43 individual patterns discussed in their paper are regularly found to be unimplementable in commercial workflow products. This means that if a real life process within a company has a requirement as is presented in the pattern, the company will have a problem correctly supporting the process using the product.

4.1.2 Verification
Eshuis [Esh02] describes workflow models to be most suited as input for Workflow Management Systems. These systems keep track of workflow instances and enable their controlled execution. He describes a technique of workflow modeling based on UML state charts. Using Temporal Logic formula’s he describes a tool that can check if propositions hold for a certain model. Using computers it is also possible to check for loops and deadlock situations by trying all possible paths. To reduce the number of paths to be checked, smart expansion strategies can be used [EW04].

5. FRAMEWORK
In the framework, the issues outlined in the previous chapter are composed into a quality framework using the principles for structuring conceptual model quality, defined by Moody [Moo05]:
- Conceptual model quality should be decomposed into a hierarchy of quality characteristics, sub characteristics and metrics.
- Single-word labels should be used for each quality characteristic and sub characteristic, using commonly-understood terms.
- Each quality characteristic and uncharacteristic should be defined using a single, concise sentence.
- Metrics should be defined for measuring each sub characteristic.
- Evaluation process: detailed procedures should be defined for conducting quality evaluations.

The evaluation process is outside the scope of this paper. The remaining 4 principles can be used as a guideline to formulate the framework. The framework is based on the articles introduced in the previous sections and features found in the 6 modeling tools. We can separate the evaluation of a modeling language from its tools, but usually, workflow languages are unique to their tools [RHA+06].
Below shows the division detailing which characteristics are relevant for just languages and which are relevant for tools implementing them:

- **Language**
  - Semantics
  - Comprehensibility

- **Tool**
  - Comprehensibility
  - Validation
  - Optimization
  - Verification
  - Implementation

The framework syntax is derived from the principles for structuring conceptual model quality [Moo05]:

- **Characteristic**
  - Sub-Characteristic description
    - Metric

The framework can be found in Appendix A.

### 6. FRAMEWORK APPLICATION

In this section, the tools and languages which they are based on are introduced. These tools have been evaluated using the framework and the results are presented.

#### 6.1 Languages

Most workflow modeling tools used in practice are based on 3 popular language definitions, some tools have extended the functionality of these concepts, while others do not fully implement certain language specifications [RHA+06a]:

- **Flowcharts**
  - Flowcharts are cause and effect diagrams which were introduced in the ANSI 3.5-1970 Flowchart Standard, they were first introduced for use in information processing in computer science. The standard was not very successful at first used only in computer science education. [Cha79] Presently, flowcharts and its derivates are very common in modeling and diagram tools.

- **BPMN**
  - A workflow modeling standard specifically designed to be more usable by all stakeholders in a project, very flexible, but lacking in certain areas and therefore unsuited to be a final specification of a workflow on it’s own, because it lacks the ability to model complex structures. [WAD+06]

- **UML 2.0 Activity Diagram**
  - Most suited for processes which are in whole part of one system. Mostly equal to BPMN in terms modeling capabilities according to Wohed et al. [WAD+06].
  - Both UML and BPMN are standards being developed and advocated by the Object Management Group.
  - Additionly, the scientific community seems to also use Petri-nets in various formats, YAWL is based on this.

#### 6.2 Workflow Modeling Tools

In this section, the six workflow modeling tools to which the framework has been applied are introduced. We use a custom workflow model (Created by Egbert-Jan Hollemann) from practice which is the first figure in Appendix A. It has been used as input to create the other models using the six tools to which the framework is applied:

- **ARIS**
  - The Aris Toolset [SN00] is sold and developed by IDS-Scheer. Aris is very popular for use in large organizations. It supports lots of different model types, and comes complete with publishing tools and various more simplified models which can be linked to Workflows and other process definitions in order to create a hierarchical model of the organization from its business goals to its IT-infrastructure [SN00].

- **BiZZdesigner**
  - BiZZdesigner, based on the AMBER business process definition [EJL+99] sold by BiZZdesign. This tool is relatively new and mostly used in the Netherlands. It can be used for targeted process improvement as well as application development.

- **YAWL**
  - Yet Another Workflow Language, an open-source initiative, which aims to be able to implement all patterns presented by the Business Process Management Center [RHA+06]. Like BPEL, this modeling language is targeted at web services and Application Workflow. It is less useful for business processes and BPR.

- **IBM Rational Modeler**
  - “IBM Rational Software Modeler, (RSM) made by IBM’s Rational Software division, is a robust Unified Modeling Language UML™ 2.0-based visual modeling and design tool. Rational Software Modeler is built on the Eclipse open-source software framework and includes capabilities focused on visual modeling and model-driven development (MDD) with the UML for creating resilient, thought-out applications and web services.” [Wik07]

- **MicrograFX Flowchart**
  - This older version of Flowcharte r uses flowcharts with optional swimminglanes. It’s primary function is to specify business processes and it can show live data (imported from external data sources) in diagrams.

- **iGraFX Process 2007**
  - Process 2007 descends from MicrograFX Flowcharter and is based on BPMN. It is often used by companies to perform process improvements like six sigma. Like Aris it scales up to enterprise level, with a model repository, management- and collaboration tools.

  These tools are a good examples of workflow tools used in practice (Aris, BiZZDesign, Rational Modeler, Flowcharter) based and more scientific solutions which are being developed to support the next generation of workflow tools which can handle more complex processes. The languages on which these tools are based are well described in science. There are a lot more tools around which can be evaluated using this framework. Findings per tool can be found in table 1.

#### 6.3 Application

To test these tools, a real business workflow taken from an already created model was implemented in all tools primary workflow diagrams. Most tools support more than one kind of diagram, Aris is most versatile, supporting a whole range of models including BPMN and UML 2.0.
Using the experience gained by creating the diagrams and by reviewing the functionality of the tools the framework (Appendix A) has been applied to them. The results have been summarized in table 1. The application of the framework on the iGrafx tool has been done by Egbert-Jan Holleman, who has created the original model using iGrafx Flowcharter's predecessor, Micrografx Flowcharter, and has more experience in workflow specification than the author. Using his comments, the framework has been further clarified by including some examples.

Table 1. Evaluation result according to framework metrics (+: positive, ±: not fully, -: negative, ?: not tested)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Aris</th>
<th>Bizdes</th>
<th>iGrafx</th>
<th>Flowcharter</th>
<th>Rational</th>
<th>YAWL</th>
</tr>
</thead>
<tbody>
<tr>
<td>How well does the workflow modeling language score on (relevant) patterns presented in [RHA+06a].</td>
<td>±  ?</td>
<td>+  +  +  ±  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the modeling language include all variables needed to use the model(s) for its intended purpose?</td>
<td>+  ±  +  +  ±  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>When multiple people draw up a model for the same domain, how many differences occur.</td>
<td>±  ?</td>
<td>±  +  +  +  ±  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How many graphics communications concepts does the language include by default?</td>
<td>+  ±  ±  ±  +  +  -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are there modeling objects which have a high resemblance, yet represent very different concepts?</td>
<td>+  +  +  +  +  +  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the tool/language support the creation of a hierarchy of models, or meta-models?</td>
<td>+  +  ±  ±  ±  ±  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the language allow and does the tool support variable emphasis on different modeling objects?</td>
<td>±  ±  +  +  +  +  +  +  -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the language include meta-models and/or does the tool support meta-views?</td>
<td>±  +  ±  +  -  -  +  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>When seeing a model of a domain, can an average person identify concepts without prior instruction?</td>
<td>+  +  ±  +  ±  ±  +  +  -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are multiple similar properties of objects being communicated? (E.g. same process types have the same color, grouping of processes according to division or product, objects being placed together in larger boxes etc.)</td>
<td>±  +  +  ±  -  -  -  -  -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do objects have a shape which clearly relates to the real-world item, and/or are objects clearly labeled?</td>
<td>±  +  +  +  +  +  +  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of elements</td>
<td>+  ±  ±  +  +  -  -  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can the tool walk a model user through the process flow?</td>
<td>-  -  +  +  +  +  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the language include a clear sequence of steps to be followed and/or is this guided</td>
<td>-  -  ±  +  ±  ±  -  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the tool allow for object representations to be changed?</td>
<td>+  ±  ±  +  ±  -  +  +  -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>After inputting a model and resources, can the tool simulate the workflow operation including use of resources?</td>
<td>+  +  +  ±  ±  ±  +  -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the tool advise on resource deployment?</td>
<td>±  +  ±  -  -  -  -  -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the tool offer suggestions to change processes to fix problems like deadlock?</td>
<td>-  -  ±  -  -  ±  -  -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can the tool check the correct use of modeling objects?</td>
<td>+  +  ±  +  +  +  -  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can the tool check for consistency of definitions across models?</td>
<td>+  +  ±  ±  -  -  -  -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the tool support a method to enter logical propositions and test them against a model?</td>
<td>+  ±  ±  +  +  +  +  +  -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Does the tool support a deadlock/loop checking?</td>
<td>+  +  ±  ±  +  +  -  -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is the language detail adequate for use as a workflow definition?</td>
<td>+  +  ±  ±  ±  ±  ±  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can the tool export the workflow definition in the input language used by a workflow management system?</td>
<td>+  +  ±  ±  ±  ±  ±  +</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can the tool export a BPEL (or BPEL4WS) specification for use as a starting point for an automated service?</td>
<td>+  +  ±  ±  ±  ±  ±  ±</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Can the workflow definition be used as a starting point for development applications?</td>
<td>-  -  ±  -  -  -  +  -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7. CONCLUSION
7.1 Summary of Application Results

Table 2. Overall framework application results per tool

<table>
<thead>
<tr>
<th>Tool</th>
<th>passed</th>
<th>partly</th>
<th>Failed</th>
<th>unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aris</td>
<td>17</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
### 7.1.1 Aris

The Aris toolset, produced by IDS-Scheer clearly has the best credentials for any application. Not only does it support EPC’s for workflow modeling, but other modeling languages like UML and BPMN are also supported. A specific model for various levels of detail and applications can be created and linked to other models to keep them synced. Using output modules, modeled domains can be used in process aware applications like SAP ERP systems and BPEL software environments. It is the clear overall winner. Aris is a very complete solution, but to use all it’s functions within an organization requires effort and expertise in modeling.

#### 7.1.2 Bizzdesigner

Bizzdesigner brings a simple to understand modeling language, consisting of a small number of modeling elements which can be easily organized into individual process blocks and subprocesses. The tool is quite capable of modeling a lot of processes, however, if it’s language definition can support more complex processes remains to be seen. More compatibility with development tools (other than BPEL) seems to be lacking. For BPR it seems adequate, although tools like Aris and iGrafx Process feel more solid in this regard.

#### 7.1.3 iGrafx Process

Process is a great BPM modeling tool and is strong when it comes to supporting BPR efforts, however, application development seems to be of less importance and it is unclear how modeling efforts can be turned into a starting point for application development.

#### 7.1.4 Micrografx Flowcharter

A clear predecessor of Process above, less refined than Process, but with the same strengths and weaknesses.

#### 7.1.5 Rational Modeler

IBM’s Rational Modeler is clearly the modeling tool for IBM’s Application development branch. Powerful intergration with software development tools leave no room for BPR facilities. The process should already be mature and set in stone when this tool comes to play.

#### 7.1.6 Visio 2007

Microsoft Visio 2007 remains the weapon of choice for drawing BPR and application development models. Vision feels very robust and allows for quick and clean model creation. The lack of language or method support is both a strength and a weakness. Especially for application development this can lead to errors. Some workflow modeling tools can take Visio models as a starting point (iGrafx Process for instance). Since Visio is very accessible for people to use, it can be very useful in BPR efforts involving employees.

#### 7.1.7 YAWL

A scientific approach to creating a flexible modeling language which can model complex services. It is not designed for use by non experts and therefore not suited for BPR efforts. It can however become useful for workflow modeling efforts which include concepts which are to complex for other languages/tools to handle correctly. Right now it is not yet production ready.

### 7.2 Conclusion

Workflow Technology has been around quite long. With the rise of the SOA principle, it has become increasingly important. However, the market for workflow software is very fragmented. There are quite some differences between various offerings and customers should be aware of that. There is both a difference in functionality [RHA+06], [GMS01] as well as the importance of making sure members of the organization can be involved in the change process and the process definition. Moody [Moo06] identifies factors which can help communicate the content of models more effectively and thus make them more accessible for members of the organization.

The framework presented in this paper is meant to create awareness about what issues affect modeling tools. If a process just needs workflow management system support, perhaps the comprehensibility section of the framework is less of an issue. If organizational processes are troubled, and need reengineering, perhaps this section is more important.

By looking at the framework results we can tell that Aris has the best credentials for both application development and BPR efforts. However, the support for multiple model types and classes and the ability to check consistency will create a lot of bureaucratic overhead to coordinate ever growing and changing sets of models if the system is fully used. Bizzdesigner offers a lot of useful functionality to use in smaller companies or in (troubled) parts of larger companies. The behavioral model method used by Bizzdesigner is clearly more abstract and less comprehensible than the standard EPC in Aris. This will make it harder to keep non-technical employees involved in the process of workflow specification/evaluation. The YAWL toolset seems even less accessible to a non-technical user. The Flowcharter (And Process) tools provide a good basis for the analysis of organizational processes. IBM’s Rational Modeler is better suited to support the creation of process aware software applications to support established business processes.

Workflow models are on the forefront of both Application Development and Business Process Development. Workflow modeling languages and tools therefore should strive to be ‘bi-lingual’ as to provide a platform for organizations to optimize their processes using information technology as well as process improvement.

### REFERENCES


[Cha79] Chapin, N. Full report of the Flowchart Committee on ANS Standard X3.5-1970 ACM SIGPLAN NOTICES Volume 14, Issue 3 p 16-27 (March 1979) ACM New York USA


APPENDIX A: WORKFLOW SOLUTION EVALUATION FRAMEWORK

- **Semantics**
  - Can the language express patterns and concepts relevant to the business domain?
    - How well does the workflow modeling language score on (relevant) patterns presented in [RHA+06a] and/or does the language encompass the aspects defined by the workflow management coalition as presented in [RHA+06b].
  - Expressiveness: do the models produced with the method provide enough detail for their intended use?
    - Does the modeling language include all variables needed to use the model(s) for its intended purpose? [homrei00]
  - Arbitrariness: Does the modeling concept allow a lot of freedom in how to express certain constructs, or are semantics strict and enforced by tools?
    - When multiple people draw up a model for the same domain, how many differences occur. [homrei00]

- **Comprehensibility**
  - Does the modeling language use multiple (or all) graphics communication concepts? (Shape, Color, Size, Value, Orientation, Texture)
    - How many graphics communications concepts does the language include by default? [Moo06]
  - Are different modeling objects easily recognizable?
    - Are there modeling objects which have a high resemblance, yet represent very different concepts? [Moo06]
  - Is it possible to subdivide models into smaller interrelated parts?
    - Does the tool/language support the creation of a hierarchy of models, or meta-models? [Moo06]
  - Does the method allow for certain elements to be emphasized (color, size etc.)?
    - Does the language allow and does the tool support variable emphasis on different modeling objects? [Moo06]
  - Is the interrelation between models easy to identify (for instance by overview models etc.)
    - Does the language include meta-models and/or does the tool support meta-views? [Moo06]
  - Is the meaning of model elements easily understandable, even without instruction? [Moo06]
    - When seeing a model of a domain, can an average person identify concepts without prior instruction? [Moo06]
  - How well are modeling objects organized according to property?
    - Are multiple similar properties of objects being communicated? (E.g. same process types have the same color, grouping of processes according to division or product, objects being placed together in larger boxes etc.) [Moo06]
  - How clearly do modeling objects relate to the concepts they represent in the subject domain?
    - Do objects have a shape which clearly relates to the real-world item, and/or are objects clearly labeled? [Moo06]
  - How many unique graphical elements does the method have?
    - Number of elements [Moo06]

- **Validation**
  - Does the tool support simulation? [BRU00]
    - Can the tool walk a model user through the process flow?
  - Does the tool/language provide a method to guide the modeling process? [Gia01]
    - Does the language include a clear sequence of steps to be followed and/or enforce a sequence of analysis.
  - Can the representation of concepts within the tool/language be altered to make it more realistic for a certain domain?
    - Does the tool allow for object representations to be changed?

- **Optimization**
  - Does the tool support quantified simulation to find bottlenecks?
    - After inputting a model and resources, can the tool simulate the workflow operation including use of resources? [Gia01],[Aa04]
  - Does the tool support automatic process change suggestions based on simulation outcome?
    - Does the tool advise on resource deployment? [Aa04]
  - Does the tool offer suggestions to change processes to fix problems like deadlock? [EW04]

- **Verification**
Can the tool check syntax and semantics?
  - Can the tool check the correct use of modeling objects?
  - Can the tool check for consistency of definitions across models?
Can the tool check if a certain logical requirement proposition is true or false?
  - Does the tool support a method to enter logical propositions and test them against a model? [Esh02] (e.g. if delivery cannot take place before payment is received the tool can check this requirement against the model and verify that it cannot occur.)
Can the tool check for deadlock or loop possibilities?
  - Does the tool support a deadlock/loop checking? [Esh02]

Implementation

Does the tool provide a workflow management system or can its process specifications be easily transferred to a suitable workflow management system?
  - Is the language detail adequate for use as a workflow definition?
  - Can the tool export the workflow definition in the input language used by a workflow management system?
Does the tool support BPEL specifications?
  - Can the tool export a BPEL (or BPEL4WS) specification for use as a starting point for an automated service?
Does the tool provide integration with software development tools such as (UML Class diagram) code generators?
  - Can the workflow definition be used as a starting point for development applications?