ILOG Components

for Business Process Management Solutions

ILOG Components for Business Process Management Solutions

White Paper

Martin Ader

Co-author

About Martin Ader:
Martin Ader set up and managed the Bull Workflow research team in 1987. The team was later transferred to Wang Software (now Eastman Software), where Martin Ader was responsible for European Workflow Marketing. In 1996, he created his own firm, Workflow & Groupware Stratégies (W&GS). Martin Ader represents W&GS at the WfMC steering committee with voting rights. Ader participates in the Workflow and Reengineering International Association (WARIA) and is WfMCs' France Chapter Chair. He has published books, articles, and papers, and participates to conferences in Asia, Europe and the United States. Martin Ader runs intensive workshops on workflow engine selection worldwide, and has had workflow consulting projects in France, Switzerland, Israel, Britain and the United States.
Workflow & Groupware Strategies - www.wngs.com

The material presented in this document is summary in nature, subject to change,
not contractual and intended for general information only and does not constitute a representation.

Table of contents

1 Foreword

1.1 Purpose and Audience

This white paper presents:

- How business users can be fully involved in the entire process life cycle through a homogeneous set of highly graphical business process definition, monitoring, and performance analysis tools.
- How business users can integrate and maintain business policies in automated processes, in ways that enable fast adaptation to market condition changes.
- How ILOG JViews for Workflow, when used in Business Process Management (BPM) or workflow products, offers more graphical and convenient tools to end-users.
- How ILOG JRules, when embedded in applications and business processes, enables business users to express business policies in a simple language.

We will explain the technological foundation of the products, the basis for their outstanding performance and features, and demonstrate the ease with which ILOG components can be integrated into many different classes of products, from Business Process Reengineering (BPR) and workflow tools to proprietary-built workflow features in large custom applications.

This white paper is intended for:

- Chief technical officers
- Product marketing managers responsible for “make or buy” decisions
- Architects and developers of BPR/BPM/workflow solutions, the people likely to use ILOG JViews and ILOG JRules
- CEOs responsible for making final decisions and covering associated risks

1.2 White Paper Structure

This paper is divided as follows:

- **“Executive Summary”** is intended for CEOs, CTOs, and engineering managers who want to rapidly understand the features and benefits that their Business Process Management (BPM) solution gains by using ILOG Java products - ILOG JViews and ILOG JRules.
- **“Typical Process Life Cycle and End-User Tools Issues”** describes briefly the typical process life cycle (which includes process definition, process monitoring, and process performance analysis phases) and the problems met by the end-users of the tools used during these phases.
- **“An Ideal Set of BPM Tools for Business Users”** describes the features of an ideal BPM tool set dedicated to business users. It includes an intuitive process modeler that enables the integration of business policies and meaningful views of executed processes displayed during enactment and analysis.
- **“Implementation Choices”** discusses the possibilities offered to ISVs and System Integrators to implement the tools described in the previous chapter. Advantages and drawbacks of the three possibilities – build from scratch, buy a turnkey solution, buy and customize components – are addressed.
- **“Enhancing the User Experience with ILOG JViews”** describes the ILOG JViews Component Suite, an ILOG product specialized in advanced graphics. It provides workflow-specific components that constitute the perfect building blocks for ISVs and System Integrators who want to rapidly build their BPM solution with high-value user interfaces for business managers.
- **“Enhancing BPM Tools’ Agility with ILOG JRules”** explains the benefits of using Business Rules technology in a BPM tool, and how ILOG JRules enables a fast and effective implementation of a rule engine.

2 Executive Summary

2.1 Workflow and BPM Challenges

> ***"Workflow is undergoing a fundamental shift as the market, vendors and products transition from a niche position to mainstream technology for enterprise application integration and e-business process automation."***
>
> ***GIGA Group***

Strong pressure to set up Internet applications quickly has changed the average duration of project life cycles from years to months. To support this pressure, more and more solutions aimed at offering online products and services are embedding workflow and rule-based technologies for the flexibility and reliability they provide:

- Workflow or process automation to manage complex business processes across organizational boundaries
- Business rules to support business policies subject to frequent adaptation due to market condition changes

The recent emphasis on Web services has sharpened the interest of companies in business processes - supported by embedded workflow technologies – as the way to model the choreography between the various components involved in a Web service.

The shift from standalone workflow products to embedded workflow technologies has given birth to Business Process Management (BPM), a new category of companies and products that integrate data, applications, and people together through a common business process to create a much more flexible framework for conducting business in the era of the Internet.

BPM vendors are now facing three challenges to get the greatest benefit from workflow technology:

- Involving business users in the business process
- Improving company responsiveness to market changes and reducing development cycles
- Optimizing processes through monitoring and performance analysis

We will now see how ILOG technology can help BPM vendors and System Integrators in winning these challenges.

2.2 Involving Business Users in the Business Process

Business users are the experts in conducting their business and must actively participate in the creation, management, and optimization of the automated processes.

Unfortunately, in most situations, the tools used at each phase of the process life cycle –definition, implementation, monitoring, and analysis – do not encourage business user participation. Indeed, these tools are often designed for technical people with notations and views not intuitive for business users. Moreover, they are usually disconnected – they may come from distinct vendors - with different models, notations, and repositories.

If tools that are too technical rapidly discourage business user collaboration, technical people also suffer from heterogeneous tools: too much training, tedious and error-prone data transfer from one tool to the other, multiple repository management.

By integrating ILOG components, BPM vendors can solve this situation and create homogeneous tools adapted to the different phases of the cycle that leverage business-user knowledge:

- A process modeler usable at definition and implementation time by business and technical people, connected to a single repository
- Representations with symbols that are meaningful to business users
- Process diagrams kept clean, even after many changes, with automatic placement of activities and transitions, and expand/collapse of sub-processes
- Process diagrams reused in monitoring views and performance analysis reports to provide visual explanations of situations
- Rule-based decision nodes expressed in a business language for describing business policies subject to frequent changes (see next section)
- Rule-based transition conditions to easily express the complexity of real-life business conditions
- Animated displays that simulate process execution

2.3 Improving Company Responsiveness to Changes

The integration of business rules in business processes magnifies the company’s ability to rapidly respond to ever-changing environments such as:

- Market condition changes that result in frequent changes in business policies
- Organization and process changes that result in complex business processes to take into account all conditions and time constraints
- Workflow execution context changes that require re-evaluation of resource allocation to reduce bottleneck risks

When combined, business processes and business rules increase the company responsiveness to these changes: The process reflects the flow of activities and data to deliver a service; the business rules reflect the business policies applied to make decisions and adapt services to market conditions.

With ILOG components, business managers express both simple and sophisticated policies from within the modeler in a readable and understandable business language. They allow dynamic activity behavior changes without any service suspension even for instances under enactment. This is done through decision nodes that are implemented as application activities parameterized with business rules.

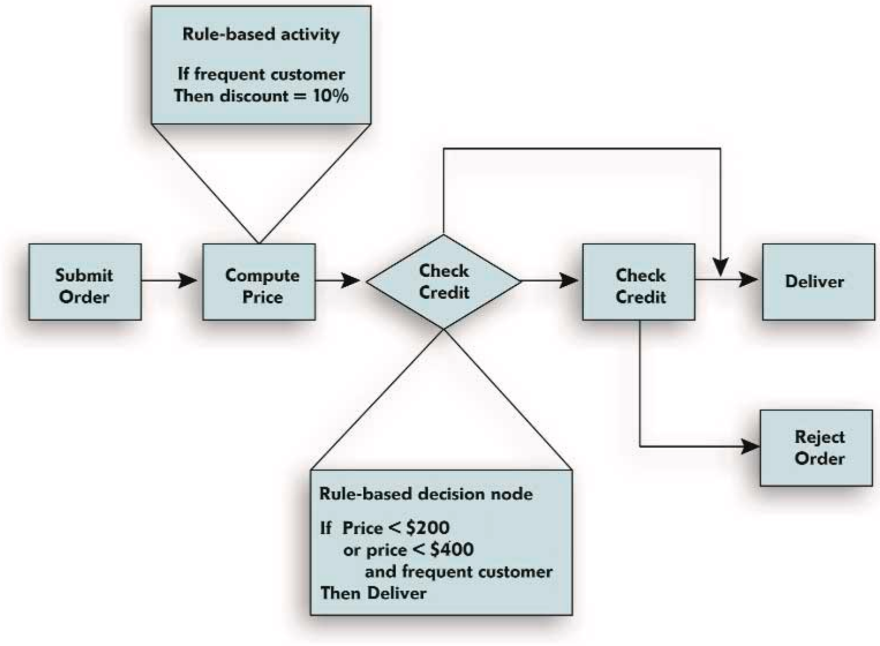


Figure 1 – Rule-based decision nodes and activities integrated in a business process

The enactment service can also be supplemented with smart and flexible assignment policies that reduce the risk of bottleneck and better use the company's resources. For instance, IT engineers can implement rules like "if an activity has been pending for more than 5 minutes, try to reassign it to another resource".

2.4 Optimizing Processes through Monitoring and Performance Analysis

> ***"From a business perspective, the greatest benefit from workflow is providing business managers with an effective tool to manage the business process (e.g., check the status of work in progress, redirect work, get performance metrics, quickly adapt processes to changing market conditions)."***
>
> ***Connie Moore, GIGA Group***

During enactment – business process execution – it is important that workflow participants and process owners have a clear understanding of their role and responsibility to act efficiently upon expected and unexpected events.

Using ILOG components, intuitive graphical views of the executed processes are displayed to help understand and solve critical situations rapidly. These views are connected to the enactment system to reflect status changes in real time.

They include:

- Tables to display process information
- Charts to display performance trends
- Process diagrams to show the actual “route”, augmented with status symbols
- Timeline views to project activities on time

The enactment system stores every event in a log database. This database constitutes a prodigious set of measurements to evaluate the performance of the company’s processes and to discover areas of improvement.

The graphical views are reused to display consolidated and detailed information. ILOG components offer the building blocks of an “executive dashboard” that the business manager uses to see the evolution of performance over time, to compare various periods, and to drill-down into classes of processes or even individual process instances.

Through intuitive views of the system, business managers and analysts easily perceive patterns and areas of improvement.

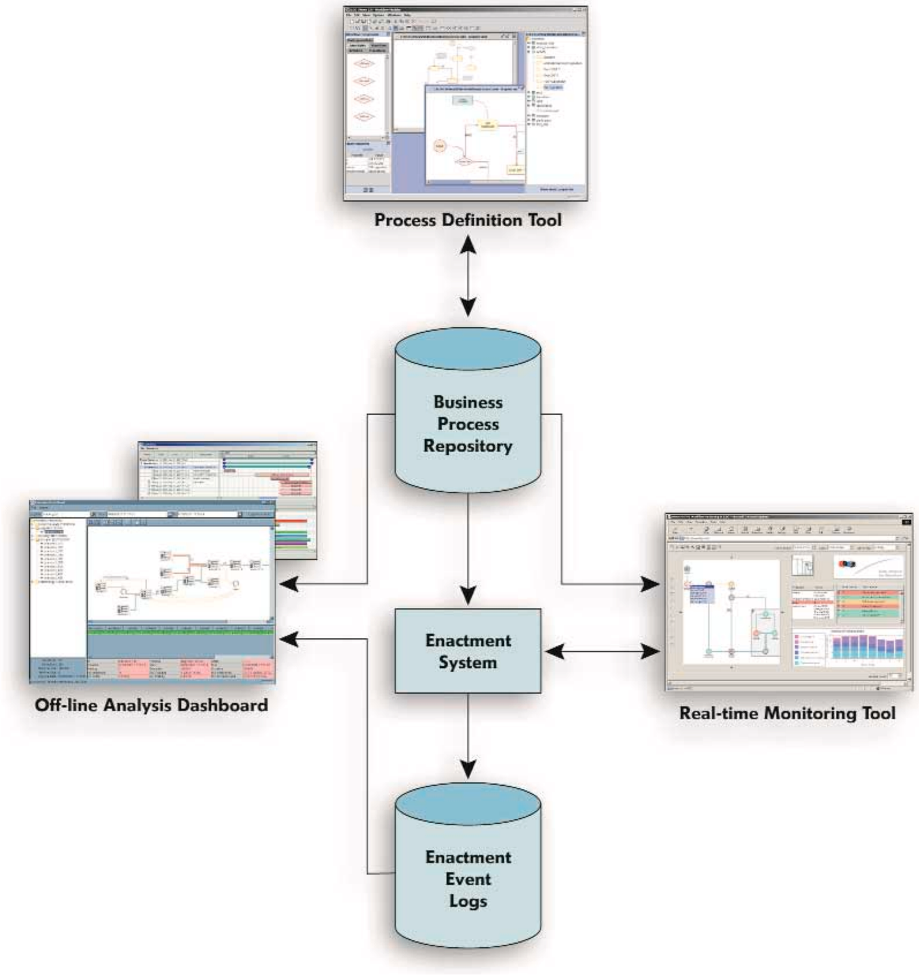

Process Definition Tool
Business Process Repository
Off-line Analysis Dashboard
Enactment System
Real-time Monitoring Tool
Enactment Event Logs

Figure 2 – Visual tools used by business users to model processes, monitor their execution, and analyse their performance.

2.5 ILOG JViews to Enhance the User Experience

ILOG JViews for Workflow provides a set of customizable Java components that help software developers *build* the effective modeling, monitoring, and analysis tools outlined in the previous section. Compliant with Workflow Management Coalition (WfMC) interfaces, these components are smoothly integrated within existing systems. Specifically, ILOG JViews for Workflow:

- Provides a ready-to-tailor visual modeling tool for defining processes
- Provides visual components for monitoring processes as diagrams, timelines, tables, and charts
- Supports complex diagramming with dynamic layout features
- Integrates with the workflow data model through XML or native workflow system APIs
- Offers the scalability required to support large, complex process definitions
- Offers services for both traditional client and thin-client deployments

By using the pre-built and pre-tested components of ILOG JViews, ISVs reduce their development time and their maintenance burden. By delivering state-of-the-art technology in an easy-to-use form, ILOG JViews empowers both business experts and end users.

ILOG JViews fully supports Java 2D and Swing. Applications developed with JViews run either on the client (Application, Applet) or on the server (Servlets, EBJ) with thin-client deployment (DHTML, SVG).

> ***"ILOG JViews for Workflow enables us to develop one of the most user-friendly process modelers on the market in a record time of two months. In a snap, our customers can adjust their business processes with this graphical modeler, giving them a tremendous competitive advantage."***
> ***Thorsten Schäfer, CTO, abaXX Technology***

2.6 Reaching Agility with ILOG JRules

ILOG JRules is a rich and flexible product enabling software developers to create applications that can be maintained with minimal effort. It allows developers to combine rule-based and object-oriented programming to add business rules to new and existing applications.

ILOG JRules includes:

- A high performance Java **rule engine**.
- A full-featured **rule language** with support for business rules. It uses domain terms so that business user expresses rules in a natural language.
- A **Rule Kit** — a comprehensive set of tools supporting the development of business rule applications: editor, Web editor, debugger, interpreter, compiler, and profiler.

ILOG JRules leverages all the unique strengths of Java, such as platform independence, multithreading, interfaces, and introspection. ILOG JRules for Java fully supports JDK v 1.2.2 through J2SDK v 1.3.1, Enterprise JavaBeans 1.1, JDBC 1.2, Servlets 2.2, J2SE, and J2EE.

> ***"Rule Based Systems (RBS) are vital to automating business functions and enabling intelligent decision-making processes."***
> ***Lynne Harvey, August 19, 1999, Patricia Seybold Group***

“High-volume e-business applications will require advanced process automation augmented by process modeling and rule-based engines”
Dale Kutnik Andrew Warzecha, Meta Group Inc 01/03/2000

2.7 Why ILOG

Selecting ILOG JViews and ILOG JRules is a strategic advantage for BPM vendors with the following benefits:

- **A solid technology** accessed through high-level components, while giving low-level control through a comprehensive API for the most demanding requirements.
- **Saving development costs** and maintenance risks by reusing components supported by ILOG professional services and R&D team.
- **From a company you can trust** with 14 years of experience in building object-oriented software components for visualization and business rules that thousands of customers worldwide use in mission-critical applications. ILOG customer support centers and consultants are available worldwide to satisfy customer demands and solve their problems.

Selecting ILOG products enables vendors to build BPM systems that place the business user in the center of the process automation life cycle.

2.8 Conclusion

Workflow technology provides the right level of reliability and flexibility for today’s enterprise applications that offer online services and products.

However, companies who will succeed in highly involving business managers in process definition, execution monitoring, and performance analysis will get the greatest benefit from embedded workflow. Indeed, they are experts in their businesses and they need effective tools to model, survey, and optimize in a non-technical way. To be adopted by business users, these tools need to:

- Offer intuitive user interfaces and effective graphical views of process models, process instances, and consolidated execution data, to model, monitor, analyze, and optimize their businesses
- Offer ways of capturing business policies subject to frequent changes due to market pressure, so that they can be managed and maintained without impacting the process flow

BPM vendors and system integrators can rely on ILOG JViews for Workflow and ILOG JRules, two pure Java components, to implement their solution rapidly, securely, with high-value services.

This concludes the executive overview. The remainder of this document further explores the technical considerations of a workflow UI, a deeper discussion of the features needed, a discussion of how to get there, and a technical description of ILOG JViews for Workflow and ILOG JRules.

3 Typical Process Life Cycle and End-User Tools Issues

This section discusses important BPM activities that require superior end-user tools, including business process modeling, business process monitoring, and performance analysis.

3.1 Typical Process Life Cycle

BPM projects generally involve the operational level (participants), middle management, and top management. All of them need to agree on the project's goals (for instance, "automation delivery") and to collaborate to define, implement, execute, and evaluate the business process.

The workflow development process is a cycle of continuous improvement. The first implementation tends to be as fast as possible and have relatively "simple" goals. Measurements collected during the execution of the first version of the process form the basis for new goals such as "reduction of delivery time", and relaunch the cycle.

Each cycle typically involves five phases:

1. Set goals
2. Process analysis, possibly using Business Process Reengineering (BPR) tools
3. Process refinements using simulation tools
4. Process definition using workflow process definition tools
5. Process measurements using workflow monitoring and performance analysis tools

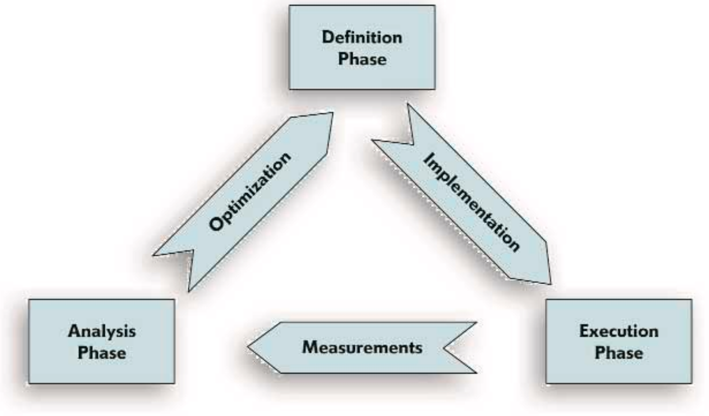


Figure 3 – The process life cycle most critical phases

The incremental nature of process development and the necessary collaboration between business and technical people impose that the tools involved be usable by all, and that they facilitate the transition from one phase to the other.

3.2 Business Process Definition

Process analysis and modeling form the most critical phases of any workflow project. They typically involve process analysts, process participants, managers, and process owners, who model the process by describing the execution order of automated or manual activities, actors or applications qualified to perform them, and transition rules between activities. Today, this task often requires the help of trained consultants.

Several meetings, interviews, prototypes, and tests are required before reaching a first working process. Process analysts drive the group, collect the information, and communicate the results as efficiently and dynamically as possible. Their task is greatly simplified when using tools that help them model process segments while running interviews and meetings.

Once modeled, IT engineers use the business descriptions that result from the analysis phase to implement the process for the actual workflow system. Ideally, the same tool should be used during both phases, the engineer supplementing the model with parameters required for the execution. Unfortunately, in many cases, the tools used for the analysis phase and at implementation are disconnected and unable to share data repositories and diagrams. Costly and error-prone data transfer is then required, considerably slowing down the development cycle and reducing motivation to incrementally refine the processes.

3.3 Business Activity Monitoring

Monitoring the actual enactment is essential for the process life cycle as it gives precise feedback on the behavior of the process instances during execution.

The enactment system – or workflow engine – creates process instances, interprets their model, and manages their execution. The engine typically routes the flow and synchronizes activities, assigns resources to activities, notifies individuals and invokes applications, and transmits data and documents to individuals and applications.

During enactment, exception situations such as overload, abnormal delays, missing goals, and process failures are notified to participants and process owners so that they can take the proper corrective actions. A deep understanding of the process context is then required to make the right decision.

In the same way as the process definition phase requires modeling tools, the enactment phase requires monitoring tools to support decision-making upon exceptions. Most workflow engines display tabular views of the process instance history that present activities executed with the participant in charge, the wait time, and the execution time. However, this is not enough to give a precise explanation of the process context.

Questions like “what is the status, which path was followed, what was done before, what will be done next, and who will do the work” would be answered using the process diagrams designed during the process definition phase. These diagrams would greatly enhance the understanding of the context, accelerate problem solving, and improve the quality of the decisions. During normal execution these diagrams would help participants understand their role in the process and enhance their motivation.

3.4 Business Performance Analysis

At enactment time, workflow engines log all events that occur in the system: process instance creation, activity creation, individual or application in charge, waiting time, execution time, and process result.

This workflow event log constitutes a prodigious database that memorizes the way a company actually executes its business. When used appropriately, it can reveal organization problems, application deficiencies, and process dysfunction that should be addressed to drastically improve the company’s execution and increase its success. From this information, corrective actions can be taken, such as:

- Better training to reduce activity execution time
- Change role assignments to minimize bottlenecks
- Increase the priority of the most critical processes
- Change the process definition to simplify the process

The workflow logs are usable by each level of the organization. Typically, participants at the operational level are interested in detailed information about processes and agreed-upon performance objectives. Middle management needs comparisons of actual results, previous periods, and planned objectives. Top management needs summary results that provide support for making strategic decisions.

Data warehouse, data mining, and business intelligence tools are often used to analyze the business process performance. Although these tools are good at statistics and classical reports using tables and charts, they totally ignore the context of the data: business processes. Consequently, process diagrams and timelines that can be so meaningful for process owners, process analysts, and even participants, cannot be used to project statistical information about execution.

3.5 Consistency Across Tools

We have seen that the whole cycle is extremely complex when not properly assisted by tools that help:

- Analysts and business users model and communicate process models
- Process owners and participants fix problems during execution
- All levels of the organization analyze the business execution to detect deficiencies and improve the company's success

Unfortunately, today's tools cover only one or two phases of the cycle and the successive use of different tools can produce different models and successive transcriptions between them. This low level of interoperability between tools does not promote collaboration between people. Rapidly, only IT engineers really master all the tools and business experts are left aside.

We did not mention business policies in the previous sections. Indeed, although they are crucial for developing agile applications, integrating business rules in business processes is still a rare feature in BPM tools.

New BPM solutions must acknowledge the incremental nature of these projects and propose tools that can smoothly cooperate and work on common data models and data repositories, and can share common visual representations.

4 An Ideal Set of BPM Tools for Business Users

BPM vendors are facing three challenges to get the greatest benefit from workflow technology:

- Involving business users in the business process
- Improving company responsiveness to changes (in particular market conditions)
- Optimizing processes through monitoring and performance analysis

Important efforts must be made so that business users become major actors in process definition, monitoring, and process optimization through performance analysis. Tools must be intuitive to be adopted by users and also to limit training, powerful to express business processes and policies, and meaningful to provoke insight and fast understanding.

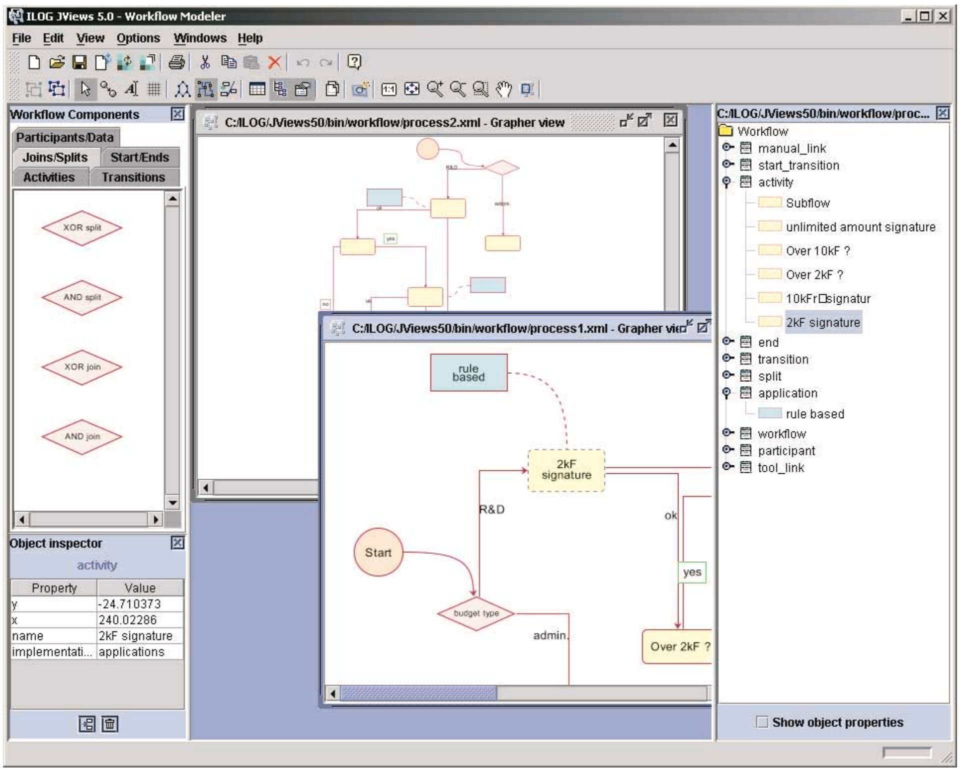
This chapter proposes features that BPM tools must provide to business users to improve their involvement in the process life cycle. These features are mainly related to the tool's user interface and the expression of market-dependent policies through business rules.

4.1 Modeling Features

The challenge of providing tools that must be used by business managers (and not only workflow experts) places tremendous demands on modeling tools.

They help visualize the process under construction and dynamically improve it. Two conditions are mandatory for the tool to be truly effective:

- The modeling formalism used by the tool must be accepted and easily understood by all the participants throughout the analysis phase
- Business policies and complex conditions must be easy to define
- The ease of use of the tool must be such that the analyst can translate “in real time” the decisions of participants into a meaningful model



Property	Value
y	-24.710373
x	240.02286
name	2kF signature
implementati...	applications

Figure 4 – A business process modeler built with ILOG JViews for Workflow

4.1.1 Basic Design Features

Although available in most drawing software, the following design tools are mandatory:

- Predefined palettes with symbols that represent workflow tasks and participants
- Icons and graphic elements that are easily understood and memorized by all participants of the process definition phase
- Connection tools for linking workflow elements together
- Drag-and-drop editing
- Property editors
- A complete set of editing tools, including selection, navigation, print, and cut and paste.

4.1.2 Advanced Design Features

These features are critical for a workflow graphical development tool and, unfortunately, only a few products have them all. They enable the use of the tool during the analysis while interacting with participants of the design phase. They help analysts organize the process diagrams so that they are kept clean, even after several incremental changes:

- Set of algorithms for laying out diagrams in seconds. They automatically rearrange workflow elements into the most meaningful representation.
- Process and subprocess representation with the ability to collapse a subprocess to simplify a repre-

sentation or expand it to go deeper into process details, with no limit to the number of levels. Alternatively, subprocesses can be represented in separate views.

- Process overview window to present the current working area within the entire process context and allow you to rapidly and precisely move from one area to another.
- Presentation of different process diagrams sharing the same basic objects (activities and roles) to easily reorganize different processes by moving parts from one to the other.
- Multiple views opened on different areas inside the same process to work on parts that are too distant to be shown in a same readable view.
- Dynamically hiding or showing categories of objects in the diagram. For example, at some point during the design the analyst may want to see only activities related to one department of the company.

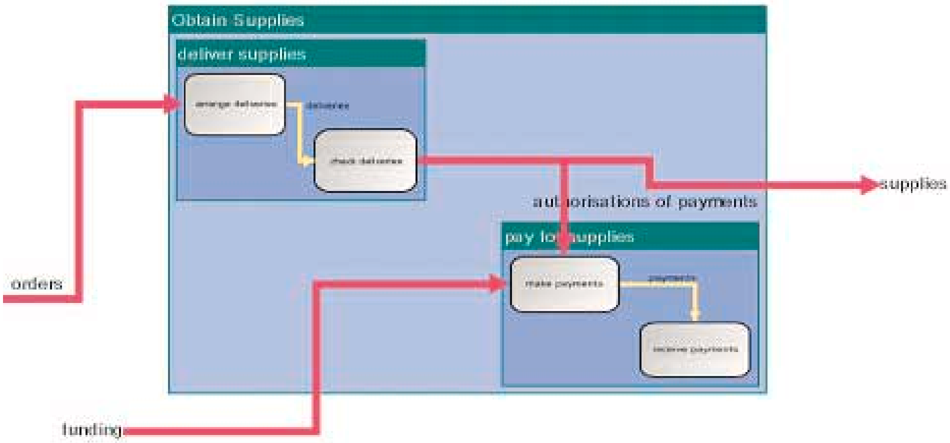


Figure 6 – "Pay for supplies" sub-process expanded and editable in place

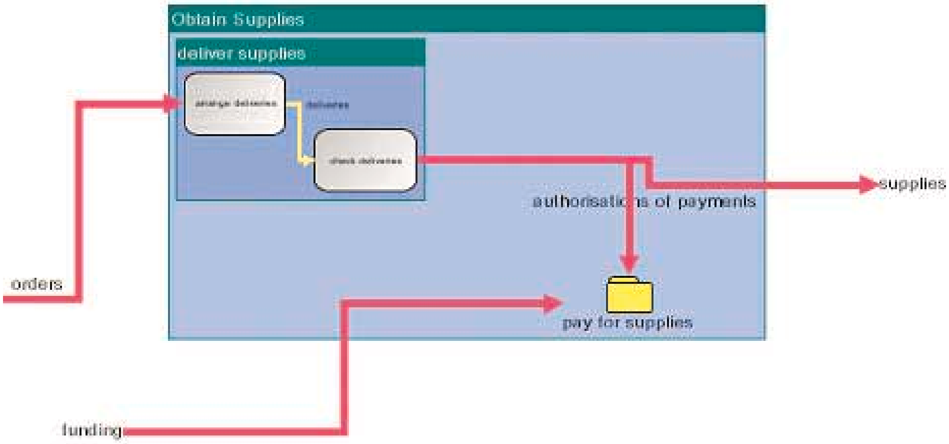


Figure 7- "Pay for supplies" sub-process collapsed

There are also a number of simulation and time features that are important. These features enable the expression of the expected waiting and execution time of activities, as well as the simulation of the processes:

- Animated graphs in which tokens “flow” through the graph to simulate process instance executions. They present an attractive view that helps users better visualize and track a process through a graph for improved performance analysis.
- Timeline-based representation in which nodes are represented as segments whose length shows their wait and execution times. Through this view, expected longest and shorted paths can be identified, and deadlines and overdue times can be adjusted to meet the expected results. It is also through this view that process simplification can be determined to reach response time goals.

4.1.3 Business Policies Features

Finally, analyst and business users must be able to easily enter business policies that are subject to frequent modifications due to market condition changes.

An ideal tool must help analysts keep business policies separated from the task flow described by the process. This is done through decision nodes that are special activities inserted in the process.

Decision nodes are implemented as applications parameterized with business rules that are created and modified using a simple language.

The rules express simple and complex policies that may modify the business process flow and data at run time.

Features include:

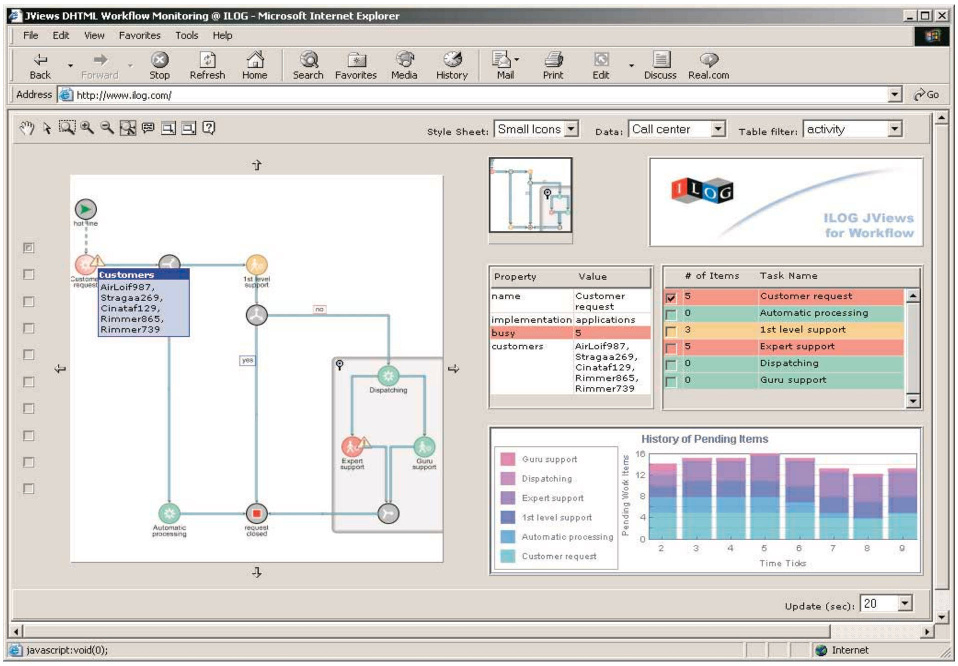
- Decision nodes (rule-based activities) inserted in the process flow.
- Direct access to the rule development environment from the modeler (by selecting a decision node) to modify existing rules or to create new ones.
- Comprehensive rule development environment usable by business users, including rule editors, interpreters, debuggers, and repositories.
- Transition conditions that can express complex real cases, including organization, time, and event conditions, with clean rules.

4.2 Activity Monitoring Features

During enactment, smart and consistent representations of process instances are required to facilitate the comprehension, interpretation, and decision-making by business participants and process owners. This is particularly true when exception situations occur that require fast reactions based on a good understanding of the whole context.

4.2.1 Basic Monitoring Features

- Tabular views of process activities under execution generally constitute the starting point of monitoring tools. Each row simply represents an activity. Columns display properties of activities such as process category, process instance to which the activity belongs, type, participant or application in charge, status, time, and so on. Sorting and filtering features can help the process owner understand the exception situation by only exposing relevant activities.
- Charts are used to represent the waiting and execution time of already executed activities for a single process instance. They can be compared to expected durations and can also help forecast the final duration of the process.



Address: http://www.ilog.com/

Style Sheet: Small Icons

Data: Call center

Table filter: activity

Customers
AirLoif987,
Stragaa269,
Cinataf129,
Rimmer865,
Rimmer739

Property	Value
name	Customer request
implementation	applications
busy	5
customers	AirLoif987, Stragaa269, Cinataf129, Rimmer865, Rimmer739

	# of Items	Task Name
☑	5	Customer request
□	0	Automatic processing
□	3	1st level support
□	5	Expert support
□	0	Dispatching
□	0	Guru support

Update (sec): 20

javascript:void(0);

Figure 7 – Web-based view for business process monitoring built with ILOG JViews for Workflow

4.2.2 Advanced Monitoring Features

Tables and charts alone cannot bring enough information to really help make the right decision when exceptions occur. Indeed, the most important source of information, which is the process model itself, is totally omitted in these views.

- The process diagram shows immediately the context of the problem: activities before and after, transition rules, actors, and applications. It is preferable to use the one designed at definition stage, as it is well known by process owners and most participants.
- The diagram is augmented with graphical indicators that represent status of activities, individuals in charge, waiting times, and execution times. Alarm symbols and colors draw attention to times exceeding the expected ones. A combination of icons, symbols, colors, and animation superimposed onto the process diagram summarizes the situation at a glance.

During enactment, the projection of information over time also provides useful contextual information:

- Timeline views represent the executed and remaining activities along a time axis. This view presents the process instance history and future, and simulates the time in which the process might be completed (for example, to answer a customer query on a delivery delay). It assists the process owner in pinpointing the main causes of delays.
- Schedule charts represent the resources activities and may help in identifying overloads and reassigning tasks to more available resources.

4.3 Performance Analysis Features

During the enactment, all events have been registered into a database (log file) that serves as a data mine for measuring the company performance and finding areas that require improvements.

The monitoring features mentioned are in fact a subset of those needed for performance analysis. The real difference between monitoring and analysis is in the user approach: the user starts reading global indicators that summarize the performance of an on-line service, a product, an individual, or a department. When problems are detected, such as exceeded average time, high error rates, or missing goals, the user tries to understand causes by looking at more detailed measurements. The usual scenario is thus typically a top-down research, from aggregated information and trends to fine details and models.

For example, analysis can be done during a short period of time (10 minutes to one hour) to detect deviations in quality standards:

- Unexpected creation rate for process instances, which might reflect many different aspects related to serving customers
- Unexpected waiting time for activities, which can indicate user overload
- Excessive execution time for activities, which can reveal a technical problem in executing the activity

Monitoring the increase or decrease of the above indicators by computing a trend indicator from the data derived from previous time-based analysis enables the prediction of possible alarm situations.

Performance analysis tools are sometimes called “Executive dashboards”. They offer high-level and detailed views of the business processes, navigation, and query capabilities, which are described in the next sections.

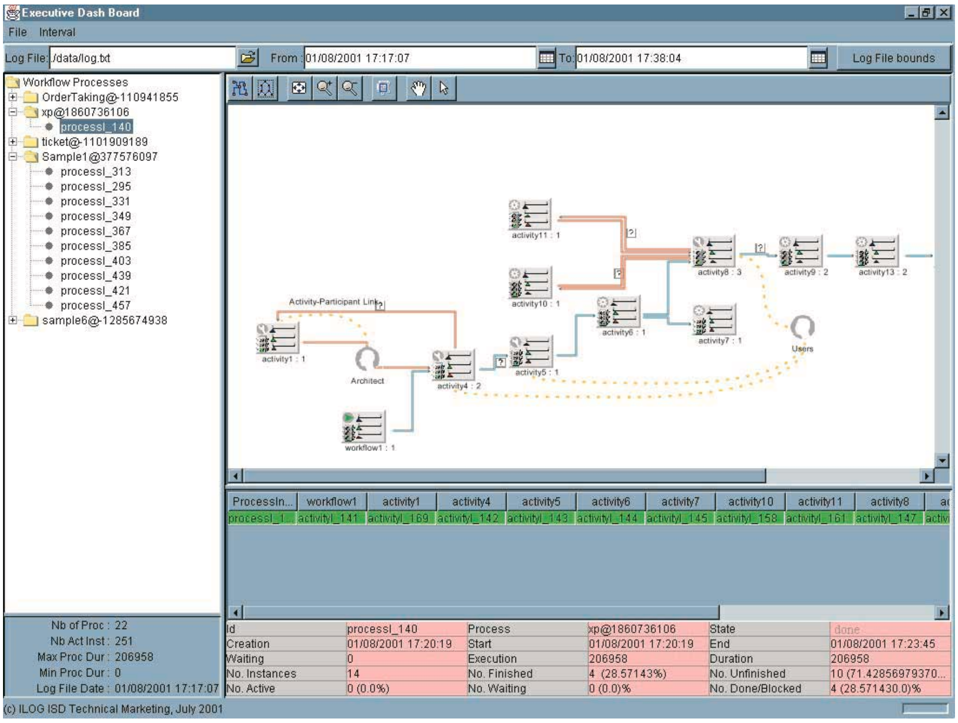
4.3.1 Basic Performance Analysis Features

- Tabular views are typically used to show aggregated data such as average time of delivery per product type. They are also used to show the full details of process instance execution when it comes to searching for particular problems.
- Radar, bar, pie, and plot charts are very efficient to display trends and detect future problems. Participants can visualize their task performance over time. By projecting the average duration of all activities for a process, middle management can see which activities consume time and should be optimized. Top management can see if the execution of the company business processes is in line with the goals and can forecast future results.

4.3.2 Advanced Performance Analysis Features

In the same way as for monitoring tools, it is crucial to supplement the above views with workflow-aware views. This is where business intelligence, data mining, and data warehousing tools become weak. We basically find the same representations as for monitoring, but instead of showing individual process instances only, the indicators also display statistics.

- Process diagrams augmented with symbols, colors, charts, and gauges showing average execution and waiting times.
- Timeline views to compare effective waiting and execution time with expected results (average or process instance details).
- Schedule charts to see how resources are actually assigned to activities during various periods.



Log File: ./data/log.txt From : 01/08/2001 17:17:07 To: 01/08/2001 17:38:04

Workflow Processes
- OrderTaking@-110941855
- xp@1860736106
  - processl_140
- ticket@-1101909189
- Sample1@377576097
  - processl_313
  - processl_295
  - processl_331
  - processl_349
  - processl_367
  - processl_385
  - processl_403
  - processl_439
  - processl_421
  - processl_457
- sample6@-1285674938

ProcessIn...	workflow1	activity1	activity4	activity5	activity6	activity7	activity10	activity11	activity8	ac
processl_1...	activityl_141	activityl_169	activityl_142	activityl_143	activityl_144	activityl_145	activityl_158	activityl_161	activityl_147	activi

Nb of Proc : 22
Nb Act Inst : 251
Max Proc Dur : 206958
Min Proc Dur : 0
Log File Date : 01/08/2001 17:17:07

Id	processl_140	Process	xp@1860736106	State	done
Creation	01/08/2001 17:20:19	Start	01/08/2001 17:20:19	End	01/08/2001 17:23:45
Waiting	0	Execution	206958	Duration	206958
No. Instances	14	No. Finished	4 (28.57143%)	No. Unfinished	10 (71.42856979370...
No. Active	0 (0.0%)	No. Waiting	0 (0.0)%	No. Done/Blocked	4 (28.571430.0)%

(c) ILOG ISD Technical Marketing, July 2001

Figure 8 – Business process dashboard showing for the selected process, the most frequently used path, the average duration and the average waiting of activities, and the rate of failure.

Navigation between summary views and detailed views becomes a crucial requirement. Obviously, the starting point and the successive views can differ according to the user position in the company. For example:

- Simple list of activities for the participants, from which they can drill-down to see the entire process to which the activity belongs as well as the other participants.
- Charts showing the evolution of performance for a business process like “customer delivery” during the quarter for top management. Each data point in the chart can represent the average process duration per day, from which it is possible to drill-down and get all details regarding the delivery processes in order to correct problems.
- Process diagrams can be overlaid with symbols and charts that show, for each activity, whether the average execution and waiting times are consistent with expected results. Middle management people can drill-down from symbols showing activities with problems, to get the complete list with participants and understand where the problem lies.

4.4 Bringing it All Together

We have seen previously that the entire project cycle is incremental and involves multiple people from operations to top management. Tools and representations that can be understood and used by all is crucial to maximize the chances of collaboration during the process definition, process execution, and process analysis.

Ideally, exactly the same models and views can be used during the whole life cycle of a process definition, from initial analysis with the end user to the monitoring and continuous improvement phases. This promotes understandability and continuity, and improves team communication and efficiency for the process of collective change management.

For example, the graph view is used at modeling time to incrementally build the representation and definition of the process. Animation in the graph view can assist in understanding and refining the process. The same view can also be used at monitoring time to provide a history of the process instance to the process owner. On the same graph, with added gauges and altered signs, workloads and alarms on process instances can be presented for selected time periods.

5 Implementation Choices

We have just described ideal features for modeling, monitoring, and performance analysis tools that are part of a complete BPM solution. We will now see how BPM vendors can effectively implement these features.

5.1 Build from Scratch

The 'build from scratch' decision is attractive as it enables BPM vendors to create tools with competitive advantages. The tools are perfectly adapted to the whole system, can smoothly cooperate, and offer consistent views along the entire cycle.

The down side of this choice is the time needed to develop advanced user interface tools:

- UI development represents often 50% of the complete system development time and is never complete: release after release, users ask for design tool improvements, look-and-feel changes, and new services.
- Must-have features like graph layout (automatic layout of diagrams) and efficient rule engines rely on extremely complex expertise and algorithms that most companies do not have in-house.

For these reasons, many companies are attracted by the option of buying turnkey applications or customizable components that plug into their core applications to reduce efforts and risks.

5.2 Buy a Turnkey Solution

The buy decision avoids development costs and risks for features that are not in the core business of the company. The three following sections discuss the options available in the three main phases of the process life cycle.

5.2.1 Modeling Tools

Business Process Reengineering (BPR) tools to be used during the analysis phase are available on the market. They provide convenient representation and editing tools but they bring up other issues:

- Additional costs related to acquiring a tool for modeling business processes.
- Maintaining two different representations of the same model at the risk of producing inconsistencies and model mismatches. BPR tools and workflow tools do not have the same process models, and processes defined with BPR tools must be migrated and adapted to the workflow tool for deployment.
- Process diagrams designed with the BPR tool are not reusable at monitoring and performance analysis time.

The consequences of using an additional BPR tool is even worse when the continuous improvement approach is adopted, which generates a new complete cycle every six months.

The convergence towards one encompassing visualization tool is important to stretch the full cycle as far as possible. This allows implementation to be completed in weeks, instead of months or years.

5.2.2 Monitoring and Performance Analysis Tools

No available products offer more than the usual tabular views of the process instances that are provided with most workflow engines.

Business Intelligence tools are usable to analyze the execution of business processes through log files, but data models and representations are disconnected from those used during the definition stage. This lack of consistent representation results in more complexity when business analysts want to improve the processes based on monitoring results.

5.3 Buy and Customize

The third choice relies on high-level components that bring the required features – ready-to-use but still customizable – and that are totally open to enable full integration with the other pieces of the system.

This choice appears to be the most promising as it combines:

- Creativity with flexible components that can be deeply customized to put added-value that makes the difference for the end-user and against the competition.
- Productivity with components that are closed to the final solution and ensure limited development cost and risk.
- Preservation of competitive advantage for pricing and features, while ensuring a complete integration with the other components of the system.

When adopting the buy-and-customize solution, choosing the right technical components and the partner that sells and supports the component becomes a strategic decision.

The components must:

- Offer a clean architecture and easy-to-learn API so that developers can rapidly master it.
- Offer extension mechanisms through documented protocols to allow full customization.
- Be designed to support incremental improvements due to customer requests and technology evolution.

Only companies with a long experience in object-oriented technologies can offer these guaranties and Java is the language and platform to go with.

ILOG is such a company, as a technology leader in object-oriented libraries and components since 1987. Its three poles of competency – visualization, optimization, and business rules – have successively given birth to lisp, C++, and Java products that are embedded into thousands of applications worldwide.

Specifically, ILOG provides two products (entirely written in Java) that provide an excellent basis for BPM vendors and integrators:

- **ILOG JViews,** with workflow-specific components and features
- **ILOG JRules,** for implementing business rules using rule technology

The next two chapters explain ILOG JViews and ILOG JRules in more detail.

6 Enhancing the User Experience with ILOG JViews

ILOG JViews, ILOG's Java visualization toolkit, is the leader in the advanced user interface market since 1997. It has introduced functionalities that enable the easy and rapid construction of BPM tools that let a single model and consistent user interface move seamlessly between the various phases of any workflow implementation cycle.

Bundled in a solution called ILOG JViews for Workflow, these new functionalities support most of the requirements described in Chapter 4.

This chapter describes ILOG JViews and more particularly ILOG JViews for Workflow, its scope, its features, and how its components can be integrated in the final application. Although the reader does not need to be a developer, this chapter is more technical than the previous ones. Executive and business readers who do not want to read it should go back to the "Executive Summary", which highlights the main principles and advantages of using ILOG JViews.

6.1 ILOG JViews Principles and Modules

The ILOG JViews Components Suite includes pre-built JavaBeans, design and editing tools, and a full-featured application program interface (API). The suite's JavaBeans – over 30 in all – enable developers to explore many of the product's key features without leaving an integrated development environment (IDE). ILOG JViews also includes design and editing tools that help define the background images that an application will use, and create custom graphics virtually without any coding. Underlying the JavaBeans and the editors is the powerful API, a set of Java class libraries that can be directly programmed to create the exact look-and-feel sought by a customer.

Written in pure Java and fully supporting Java 2D rendering capabilities, the suite accommodates a variety of code development and deployment environments. It can be used with any IDE – the included JavaBeans can be easily installed into them and immediately used – and alongside any Swing, AWT, or third-party vendor's components. Integration hooks include Scalable Vector Graphics (SVG) inputs/outputs, dedicated XML formats, and custom connectors.

Finally, for WEB applications in which non-Java clients are desired, the suite can be deployed on the server, where it can generate interactive DHTML bitmap displays or SVG, an emerging browser standard for vector graphics.

The ILOG JViews libraries are organized into several modules:

- Graphics Framework module, which offers the object-oriented layer on top of Java2D to handle graphic objects, interactions, views, and diagrams.
- Graph Layout module, which provides several algorithms that automatically position the nodes of a diagram and route its links, thereby keeping the picture understandable.
- Maps module, which enables the integration of vector and raster digital maps and the placement of application objects with latitude and longitude coordinates, using projection systems.
- Stylable Data Mapper (SDM) module, which brings together the above services in a powerful, ready-to-use component, thereby saving months of development. SDM is the central component used in ILOG JViews for Workflow to define and monitor the business processes. We will come back to it in detail.
- Gantt Chart module, which facilitates the visualization and editing of scheduling data sets through a pre-built component. The Gantt Chart is used by BPM systems to implement timeline representations.

- Charts module, which offers line, pie, bar, and many other types of charts designed to support huge data sets, even in real time. Charts are used in BPM applications to represent trends and statistical results.

6.2 ILOG JViews for Workflow Scope

ILOG JViews for Workflow, a JViews solution that is part of the component suite, is specifically designed for creating state-of-the-art user interfaces for BPM. It uses several modules of the JViews component suite and adds source-code examples that serve to illustrate the usage of JViews in BPM applications.

It provides components to present and define processes through several views:

- Diagram views for representing the activity flow through nodes connected to links
- Timeline views, in which activities are represented with bars whose length is proportional to their expected (or measured) waiting and execution times
- Table views for representing processes in a tabular way, with all their associated fields
- Chart views for representing measurements, trends, and performances

Its predefined workflow icons, smart diagramming capabilities, real-time animation of screen elements, and a complete set of editing functions enable Java developers to deliver:

- An entirely new class of advanced workflow modelers
- Executive dashboards and monitoring tools

6.3 Model-View Architecture

ILOG JViews' SDM module is a central component of ILOG JViews for Workflow as it provides the data model used to connect the various user interface screens to the underlying workflow system. The next sections explain its architecture and services.

SDM uses the Swing architecture principles of maintaining a clean separation of data from representation. With this division, only the data is considered when integrating the component, leaving no need to deal with representation objects and graphics problems. Populate the data model and add, remove, and modify the data: everything is transparently reflected in the graphical views. Similarly, every user interaction on the graphical representation is actually transmitted and performed by the data model.

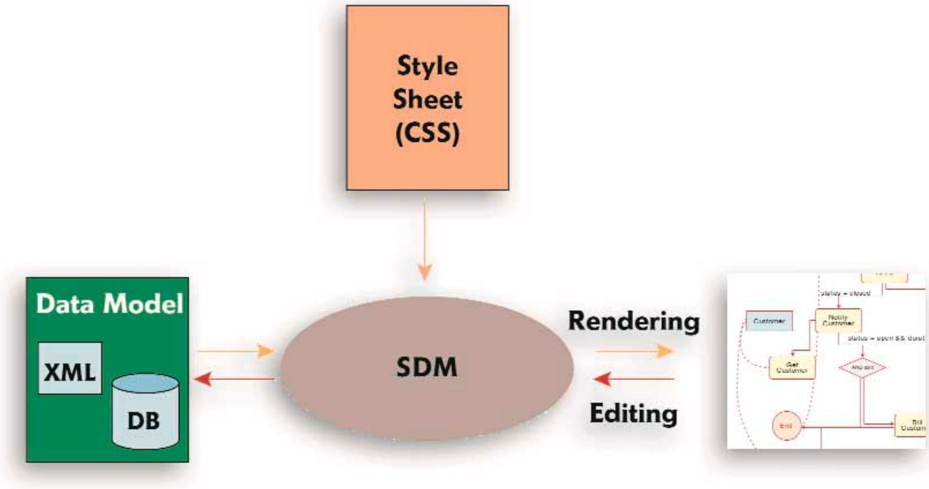


Figure 9 – SDM architecture

SDM is composed of four key elements:

- The data model connects to the application data
- The grapher displays the diagram
- The style sheet defines how the data is displayed graphically
- The engine binds the above three elements together to create the view

6.3.1 Data Model

Basically, the data model is a hierarchical graph (nodes and links). The types of data recognized by the data model include by default:

- Nodes: activity, join, split, application, data, participant, workflow, and end
- Links: manual, tool, data, subflow_reference, start_transition, and transition
- Properties that are added to nodes or links: ID, name, route predicate, kind of activity, type of join node, type of split node, kind of participant, link origin, link destination, and condition

The SDM data model declares a Java API used to manipulate the data model and react to various events coming from the workflow system or the user interface. It can be fully extended, without any coding, to take into account each and every particularity of the workflow system data – additional data, parameters, events, and so on.

Compliance with the WfMC Interface 1 ensures that the needs for such extensions will remain limited for workflow systems that comply with this standard.

XML can be used (for instance, for fast prototyping) to exchange data with the workflow repository or enactment system.

6.3.2 Rendering and Animation

The mapping between the type and property of data in the data model and its graphic representations in the various views is entirely defined in the style sheet, using the Cascading Style Sheet (CSS) language. CSS is the standard language used by Web designers to apply style to HTML pages. ILOG JViews has extended the scope of CSS to enable the styling of Java components.

Using the style sheet, the SDM engine transforms any information of the data model to graphics representations in the grapher.

When connected to live data, the changes in the data model are notified to the SDM engine, which refreshes the views immediately. The resulting animation is used to simulate the process or to reflect the enactment of a business process in real time. Animated graphic objects such as gauges, charts, and blinking alarms are also used to efficiently reflect status changes.

When other visualization components are connected to the SDM data model to provide alternative views such as tables, charts, and forms, they are all kept synchronized through the data model.

```
node.workflow {
       icon          : url(images/start.gif);
       iconPosition  : Left;
       shapeType     : Rectangle;
       shapeWidth    : 170;
       shapeHeight   : 50;
       foreground    : black;
       fillColor1    : 198,226,255; // slate grey 1
       fillStyle     : SOLID_COLOR;
}
```

Figure 10 – Example of a CSS rule describing the shape, icon, color of a "workflow" node.

6.3.3 Interaction

Ultimately, the business analyst interacts with the process diagram to edit it, to inspect its activity properties, to zoom, to pan, and to drill down. When the user adds an activity graphically, the activity is actually created in the process data model that in return notifies all the connected views. New interactions can be easily developed and integrated.

6.4 Integration, I/O

In the section about SDM data model, we have briefly seen that it can be loosely coupled to the workflow system using XML, or tightly coupled using Java API. The following sections depict these two possibilities and their advantages.

Despite the WfMC efforts, most workflow systems do not comply 100% with WfMC standards: additional data and special fields are often critical for any particular workflow system. Yet, with ILOG JViews, the connection with the standard part is straightforward and the adaptation to particularities is easy.

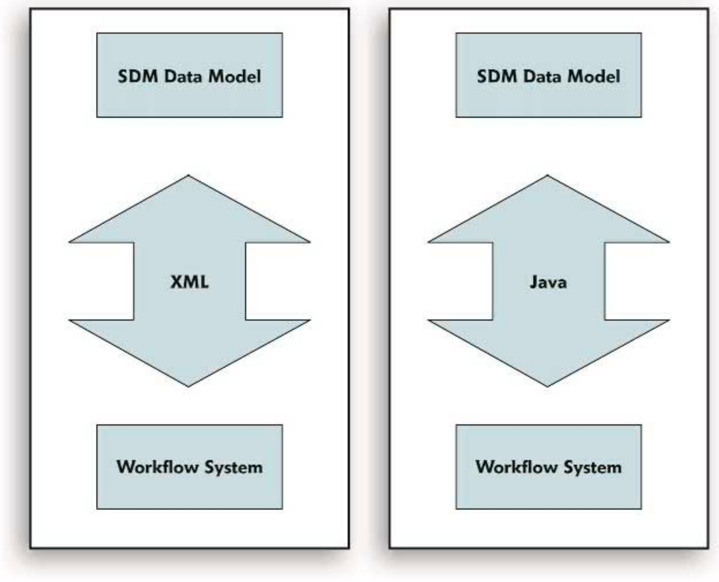


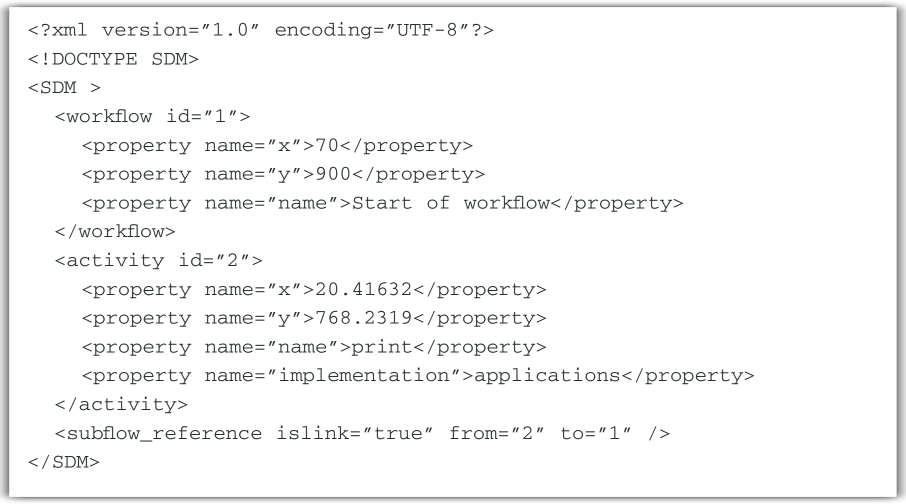
Figure 11 – SDM can exchange data with the BPM system using XML or Java.

6.4.1 Loosely Coupled Through XML

The SDM data model supports XML for loosely coupling SDM to applications. The XML format used in ILOG JViews for Workflow is inspired by the WPDL format defined by the WfMC, and can be adapted to the actual workflow system data by just adding new properties. When a data type is added to the JViews format, a new CSS rule is inserted to define the way this data is represented graphically. A few days are usually enough to adapt the XML and declare the graphics.

When the workflow system has its proper XML format, it can be declared as a new format for JViews, or transformed to the JViews format using XSL-T. A couple of weeks are enough to integrate a full XML format into SDM.

Once everything is done, the system is coupled and you have a process diagram connected to the system for either modeling processes or monitoring them.



```
<?xml version=”1.0” encoding=”UTF-8”?>
<!DOCTYPE SDM>
<SDM >
  <workflow id=”1”>
    <property name=”x”>70</property>
    <property name=”y”>900</property>
    <property name=”name”>Start of workflow</property>
  </workflow>
  <activity id=”2”>
    <property name=”x”>20.41632</property>
    <property name=”y”>768.2319</property>
    <property name=”name”>print</property>
    <property name=”implementation”>applications</property>
  </activity>
  <subflow_reference islink=”true” from=”2” to=”1” />
</SDM>
```

Figure 12 – Example of a XML document generated by SDM to implement a business process

6.4.2 Tightly Coupled Through a Java API

Full integration of the SDM component into the workflow system is possible if the workflow system offers a Java API to query and modify its objects.

The SDM abstract data model is then used as a starting point to implement the connection to the workflow system. The Java developer implements the data model methods to tell SDM how to get or set the workflow system data. For instance, the method `public java.lang.Object createNode(java.lang.String tag)` creates a new node with the property indicated by the tag. The developer implements this method using the workflow system Java API to actually create new data in the workflow system.

6.5 ILOG JViews For Workflow Features

Now that we have seen the architecture of SDM, its behavior, and the way it can be integrated, we are going to see the features that come with using SDM in ILOG JViews for Workflow.

6.5.1 Symbols for Business Users

ILOG JViews for Workflow comes with a full set of predefined symbols for representing flow charts in an intuitive way. These symbols are bound to the data model through the style sheet.

6.5.1.1 High-quality Rendering

ILOG JViews is based on Sun Microsystem’s Java 2D to offer high-quality rendering possibilities such as transparency, color gradient, shading effects, anti-aliasing, and so on. These rendering effects improve the overall quality of the process diagrams and enable a more meaningful graphical notation, which can convey a great deal of information.

6.5.1.2 Fully Customizable Look-and-Feel

The visual side of any application is critical to the adoption of the system by its users. Corporations communicate their values through images, logos, and colors, and standardize their application user interfaces with a corporate look-and-feel that projects these images.

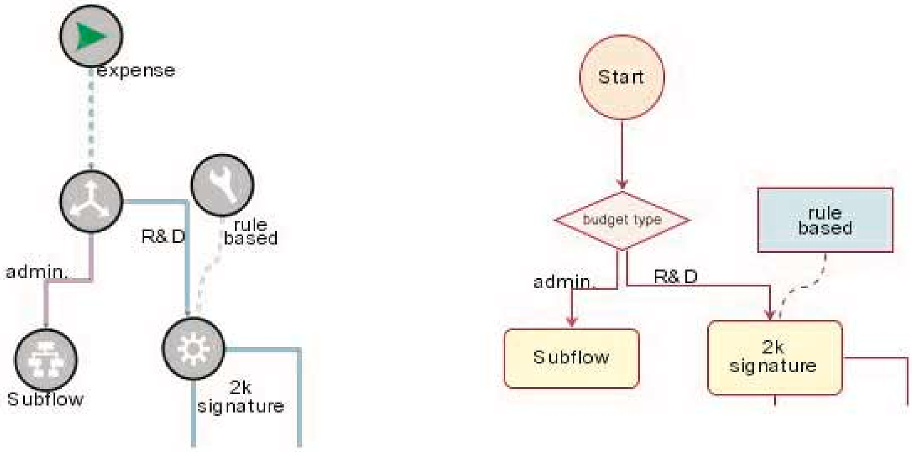


Figure 13 – By just changing the CSS file, the same business process can be rendered for business users or for developers.

Visual tools developed with ILOG JViews are fully integrated at the look-and-feel level with the other corporate applications. Through CSS, no tedious development is required: simply declare new styles to create a completely new look-and-feel. Custom CSS files can be delivered for each workflow customer and modified even by end users, if desired.

Adapting the representation to the type of user – business or technical – is a matter of changing the CSS file. Business process elements that are only meaningful for implementation can be hidden to business users.

6.5.1.3 Animated Symbols

Data types can be represented with simple icons or labels, or with combinations of objects. Activity status changes can trigger color changes, object disappearance, and gauge animation for instance.

Process simulation and monitoring tools require symbols that display numeric values with gauges, meters, or charts, with inherent animation capabilities such as blinking, rotating, and translating.

ILOG JViews provides the visual tool to graphically build any new type of symbol: this tool is called Composer.

Creating a new animated symbol involves the following steps:

- Draw the object by assembling shapes, or import drawings done with professional design tools such as Adobe Illustrator.
- Define the symbol properties used to connect the drawing to the application data.
- Associate behaviors such as rotate, blink, move, and change color to parts of the symbol and determine the property that triggers each behavior.

Wizards drive the user during this process, and the symbols are eventually stored in symbol palettes reusable across projects.

The new symbol is finally used in a process diagram by declaring a new CSS rule that maps the process object properties to the new symbol properties.

6.5.2 Graph Drawing and Automatic Layout

SDM is composed of a data model, a style sheet, an engine, and a grapher. The grapher is a graphic object container specialized to display and to interact with networks, diagrams, and anything representing nodes and links. It is optimized to support huge amounts of nodes and links while preserving high-speed refreshes and user interactions: zooming, panning, object selection, and drag-and-drop, all scalable to thousands of objects.

6.5.2.1 Graph Layout

No complex process can be efficiently defined and maintained without automatic layout functions. Models are created incrementally with multiple insertions and deletions of activities and transitions that usually result in messy diagrams requiring tedious manual repositioning to clarify. All this is avoided with layout algorithms that automatically position the nodes and route the links.

The ILOG JViews Graph Layout module, accessible through SDM and CSS, contains more than 10 layouts (including flow-specific ones) that instantly clarify even large models. Parameters let users find a good compromise between aesthetic and performance: real-time layout done while the end-user is editing, or off-line more aesthetic layouts computed once the model is edited.

Benchmarks at the end of this chapter show that results are obtained within seconds, not minutes, even with large models.

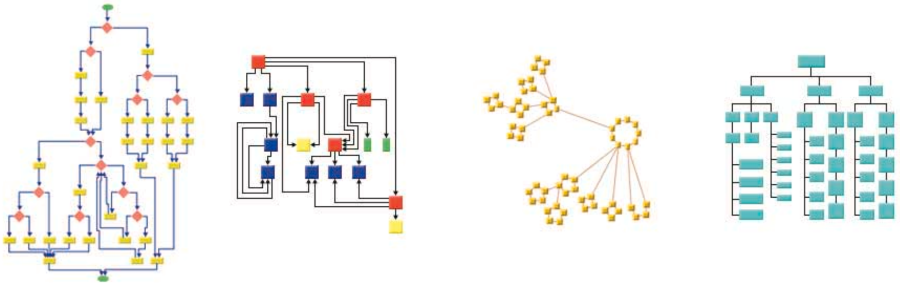


Figure 14 – Several graph layout algorithms are available to arrange virtually any diagram and keep them readable

6.5.2.2 Expand/Collapse Subprocesses

Complex processes result in intricate diagrams that are difficult to understand, maintain, and edit. Most of them are composed of subprocesses that sometimes need to be displayed entirely or as single symbols.

The expand/collapse feature provides a clean way to simplify the diagram by giving the user the ability to collapse some parts and expand others. Contrary to most diagramming tools, inter subgraph links do not stop at the subgraph boundary; they can go from a parent node to a child node.

Alternatively, entirely new views can be created to edit subprocesses.

6.5.2.3 Multiple Views and Layers

When processes are too large to be entirely displayed on a computer screen, the user can zoom in a particular region and keep an overview of the entire process. The user can also work on different areas of the process simultaneously. The multiple-views support feature of JViews acts as if several cameras were looking at different regions with various zoom levels.

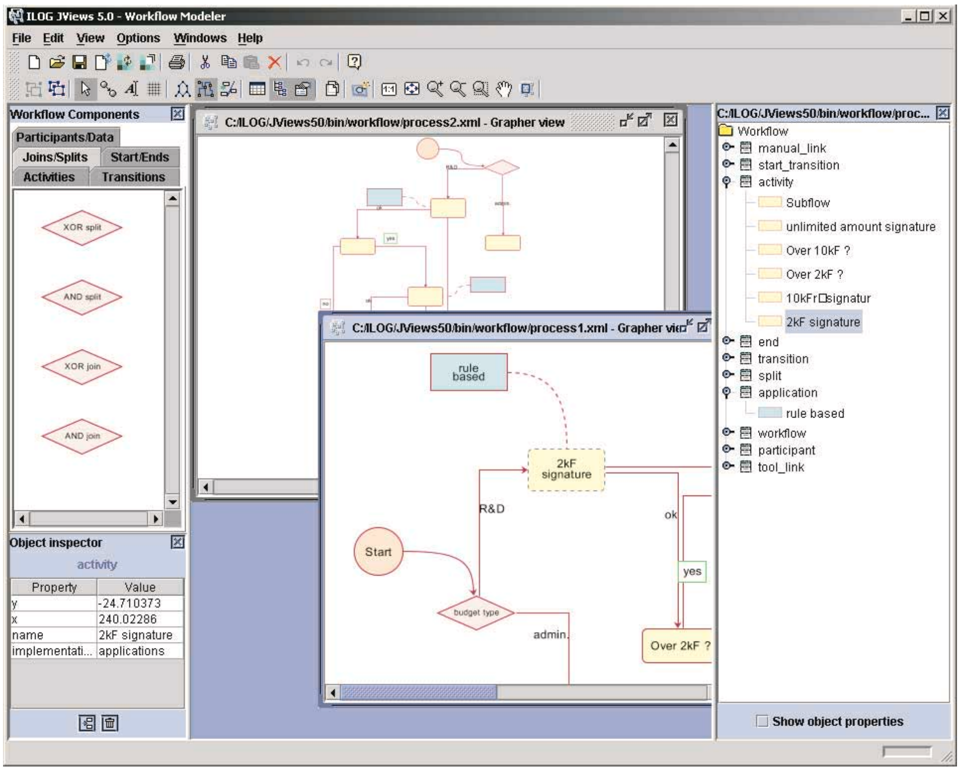
JViews also supports layers that are set to visible or invisible, and can be associated to specific application conditions. For instance, when layers are associated with company departments, the user can filter the flow chart by showing only activities related to certain departments at a time.

6.5.3 Workflow Modeler

ILOG JViews for Workflow delivers a process definition tool built with SDM that is provided with source code: the workflow modeler. The benefits of having such an important application in source code can be summarized as follows:

- Being able to deliver a complete modeler quickly. The integration can be done with XML or a Java API, as seen in the Integration section.
- Being able to enrich the current modeler, without having to start from scratch. The complete ILOG JViews and Swing API can be used to add palettes, symbols, forms, dialogs, object inspectors – everything that is required to reach the level of expectation set by the end users, and add specific value-added services that make your modeler competitive.
- Being able to reuse parts of the source code in slightly different contexts for better productivity and illustration of SDM capabilities.

The number of lines of code required to build a workflow modeler on top of SDM gives an idea of the power of SDM: 1,300 lines of code are enough to provide the modeler.



Property	Value
y	-24.710373
x	240.02286
name	2kF signature
implementati...	applications

Figure 15 – ILOG JViews modeler

6.5.3.1 Modeler Services

The workflow modeler provides the classic visual editing functions generally given with such tools, including a set of menus, drag-and-drop, selection, and palettes of workflow objects.

Other interesting services include:

- Automatic layout while editing the graph
- Collapse and extend subgraphs or parts of the graph, through the group/ungroup command
- Overview window to easily navigate in large diagrams
- Tree view of the process
- Multiple views on the same process
- Layers to filter the graphics based on some particular semantic
- Input/output XML format, specific XML formats, or directly manipulate process objects
- Dynamic look-and-feel modification by loading new CSS files

6.5.3.2 Modeler Extensions

Extension mechanisms have already been covered in describing SDM in the previous sections. Since the workflow modeler is just an application using SDM, it can be extended to better answer specific demands.

On the model side:

- The default XML format can be extended to map all the workflow system data.
- New XML formats can be added.
- The modeler can be directly connected to the workflow system by implementing a specific SDM data model. By doing this, the modeler uses the workflow system API to get, set, and modify processes, resulting in a complete integration that guaranties speed and security.

On the user interface side:

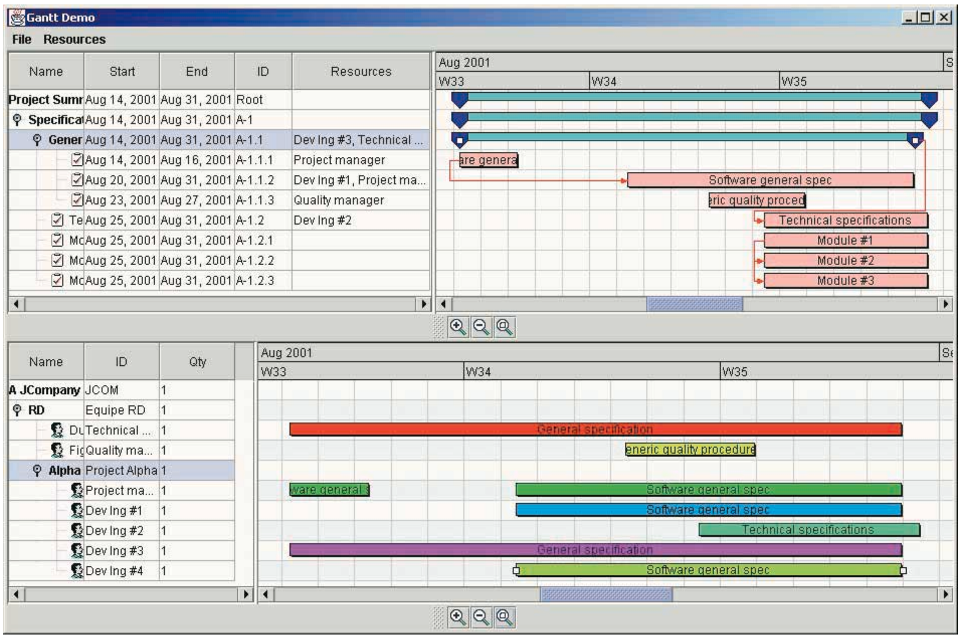
- New style sheets can be added to apply the corporate look-and-feel.
- New graphic objects can be added to represent additional information.
- New inspectors, palettes, dialogs, and interactors can be added to enhance the end-user interaction level.

Since the workflow modeler is given with all its source code, it can be used as a source of inspiration to add new functions.

6.5.4 Gantt Chart and Charts

We have seen in the previous chapters that representations other than flow charts provide meaningful information: representing activities over time and representing the evolution of performance or trends over time.

They are supported by other ILOG JViews modules built following the same architecture and extension principles as for SDM and also integrated through XML or Java.



Name	Start	End	ID	Resources
Project Sumr	Aug 14, 2001	Aug 31, 2001	Root	
Specificat	Aug 14, 2001	Aug 31, 2001	A-1	
Gener	Aug 14, 2001	Aug 31, 2001	A-1.1	Dev Ing #3, Technical ...
☑	Aug 14, 2001	Aug 16, 2001	A-1.1.1	Project manager
☑	Aug 20, 2001	Aug 31, 2001	A-1.1.2	Dev Ing #1, Project ma...
☑	Aug 23, 2001	Aug 27, 2001	A-1.1.3	Quality manager
☑ Te	Aug 25, 2001	Aug 31, 2001	A-1.2	Dev Ing #2
☑ Mc	Aug 25, 2001	Aug 31, 2001	A-1.2.1	
☑ Mc	Aug 25, 2001	Aug 31, 2001	A-1.2.2	
☑ Mc	Aug 25, 2001	Aug 31, 2001	A-1.2.3	

Name	ID	Qty
A JCompany	JCOM	1
RD	Equipe RD	1
Du	Technical ...	1
Fig	Quality ma...	1
Alpha	Project Alpha	1
	Project ma...	1
	Dev Ing #1	1
	Dev Ing #2	1
	Dev Ing #3	1
	Dev Ing #4	1

Figure 16 – Timelines implemented with Gantt charts, to map executed and planned activities along time.

6.5.4.1 Gantt Chart Features

The ILOG JViews Gantt Chart module provides complete support for viewing schedules – both resource- and task-oriented – in a Web-based display. Features include completely customizable renderings and user interactions, the ability to handle very large sets of data, and an architecture that provides a clean separation of data from the display.

Typically, scheduling information contains four data types:

- Activities - tasks to perform
- Participants - individuals, machines
- Reservations - assigning resources to activities
- Constraints - dependencies between activities (for example, “activity1 must start before activity2”)

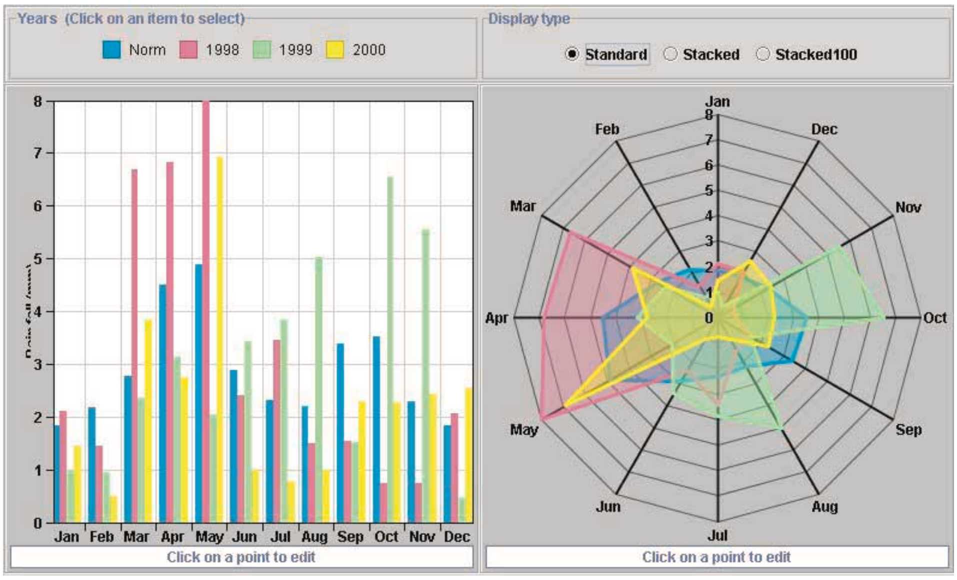
This information is displayed as:

- ***Activity-oriented chart (Gantt chart)*** indicating when tasks are scheduled, as well as constraints between tasks. When analyzing the performance of a process instance, Gantt representations are used to display when the activities of a process were actually performed (creation, execution, and termination) overlaid with expected results.
- ***Resource-oriented chart (schedule chart)*** indicating participants and their scheduled activities. The focus is then on the workflow participant for which we want to see periods of overload or periods that are particularly quiet.
- ***Table view***

Participants or activities are presented in a table derived from Swing’s JTable. Rows can be expanded or collapsed to represent subresources or subactivities.

6.5.4.2 Charts Features

The ILOG JViews Charts module offers charting capabilities for both full Java clients and thin clients. Predefined chart types are available for the most common needs, and brand new chart types can be created with the Java API. As with the other ILOG JViews modules, the clean separation between data and rendering makes it easy to implement new data models or to create new types of charts.



Years (Click on an item to select)

Display type: ● Standard ○ Stacked ○ Stacked100

Figure 17 – ILOG JViews charts

Predefined chart types include both Cartesian and polar charts:

- Scatter (data displayed with markers)
- Polyline and area, both in superimposed or stacked mode
- Bar (three representation modes: clustered, superimposed, and stacked)
- Stair (a continuous representation displaying transition between data points)
- Bubble (data displayed with bubble-looking shapes)
- Pie charts
- Radar charts
- Stock representations (including High-Low-Open-Close and Candle modes)
- HighLow (four representation modes: bar, arrow, stick, and marked stick)

The Charts module also offers many interactions so that the end-user can efficiently browse the data, zoom-in and out, edit the data, and so forth.

6.6 Web Publishing

Up to now, we have covered the use of ILOG JViews for Workflow in a power-client context: Java application and Java applet. These two ways of deploying an application are crucial when the level of interaction is important, as with a visual modeler, and when the network bandwidth is high enough, as with intranet networks.

Other than these cases, there are also many reasons for publishing process representations using the Internet: when the level of interaction between the graphics and the end-user is low and when the graphics do not have to be refreshed too often. This is the case for Web-based process documentation and monitoring, report, and analysis.

ILOG JViews components are not only embedded in Java applications and applets, but also in servlets that reside on the Web server. The ILOG JViews component is then able to receive commands from the Web clients and respond by sending back raster images (JPEG, PNG) or vector images (SVG).

6.7 Benchmarks

6.7.1 Conditions

The following test measures the performance of ILOG JViews for Workflow in the Workflow Modeler application.

The process used for the test contains 500 nodes and 500 links, which represents a typical large process.

The loading time is broken down into several phases:

- Load and process XML file to build the data model, apply the style sheet and build the graphic objects, perform the link layout, and display the graphic objects.
- Perform the automatic layout of the links.
- Optionally perform the layout of the nodes. During the process definition, the node layout is usually performed and the node positions are saved in the XML file. Therefore this phase is not required when reloading a process.
- Refresh the screen after a complete erasing (pure drawing time).

The rendering phase time depends on the complexity of the style. The more complicated the objects, the more time is required to build and render them. Advanced rendering capabilities such as color gradient, shading, transparency, and zoomable labels are also time consuming. The test uses several styles to show how the rendering phase influences performance:

- Complex style with color gradients, zoomable labels, complex shapes, and links with gradients and dotted lines.
- Intermediate style with no color gradient, same shape for each node, and no gradient for links (width 4).
- Simple style corresponding to the styles used by most BPR and workflow tools: nodes are rectangles and links are simple polylines with width 1.

Finally, the test is performed on a DELL Precision 420, Pentium III 933 Mhz, 512 Mo of RAM, with JDK 1.3.0.

6.7.2 Results

Style	*Load+display*	*Node layout*	*Link layout*	*Refresh*
Complex	3.00	1.00	0.13	0.53
Intermediate	1.30	0.50	0.12	0.20
Simple	0.50	0.40	0.10	0.10

Test: Results are expressed in seconds. The process is composed of 500 nodes and 500 links.

7 Enhancing BPM Tools´ Agility with ILOG JRules

7.1 Business Processes and Business Rules

We have seen in the previous section that ILOG JViews for Workflow is a very interesting set of Java components for developing highly visual BPM tools dedicated to business users (these tools include process modelers, real-time monitoring dashboards, and intuitive analysis views).

If these tools cover perfectly the workflow side of BPM tools – that is, the flow of data and activities through the organization – the business side is still to be implemented. Indeed, business users do not just need to model, monitor, and analyze the process as a flow of information; they also need to smoothly integrate business logic in the entire automated service with the following constraints:

- It must be expressed using languages that are natural to business users
- It must be easily modified to reflect the frequent market condition changes, strategic decisions, marketing campaigns, product promotions, and competition threats.

Because of these two constraints, the business logic cannot be implemented as traditional application code or as workflow models, which would both require a complete project life cycle at each modification.

Business Rules brings the appropriate answer:

- Expressed in a readable, understandable business rule language – rather than a programming language – they can be understood, written, and managed by business users and business analysts who are responsible for defining the business logic.
- Externalized from the application code or the business model, they can be changed independently without recompiling the application or redefining the process. Online services are adaptable and maintainable.

Once business rules are captured, expressed formally, and managed systematically, they become a powerful tool for IT departments and business users. No longer buried in computer systems in the form of code, they become accessible to non-programmers. The clarity and precision of their representation enables more effective communication and faster decision-making across the enterprise. This results in shorter process design times and shorter time to market.

> ***"We selected ILOG JRules because its flexibility allowed us to integrate the technology in record time - crucial for fast deployment - and the customizable business rule language support will allow us to offer a solution that business people can use. Customers are demanding more personalized web experiences and the ability to receive the right offer at the right time based on their need. ILOG's JRules will help us to deliver that one-to-one marketing."***
>
> ***Sam Spadafora, CEO for Chordiant Software.***

7.2 Introducing ILOG JRules Technology

ILOG JRules is a rich and flexible product enabling software developers to create applications that can be maintained with minimal effort. This section presents an overview of the distinguishing features that set ILOG JRules apart as the technological leader in business rule engines.

ILOG JRules includes a high performance Java rule engine, a full-featured rule language with support for business rules, and the Rule Kit — a comprehensive set of tools supporting the development of business rule applications.

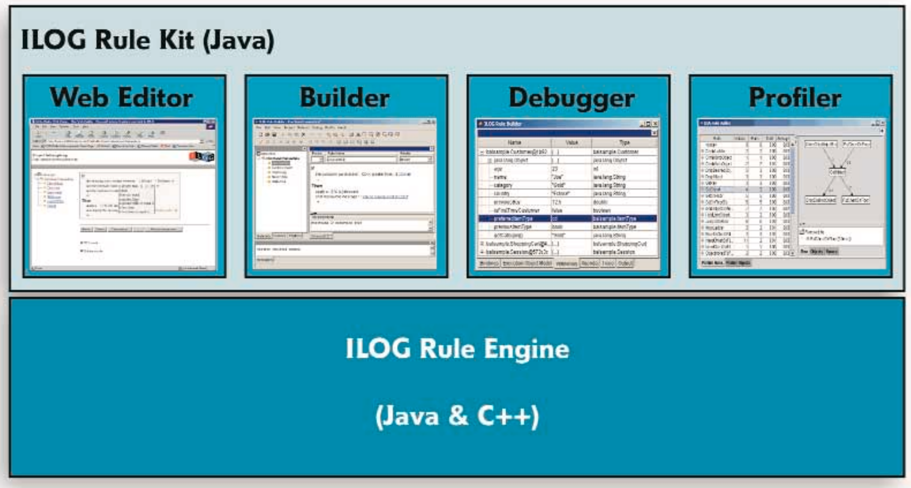


Figure 18 – ILOG JRules components

ILOG JRules leverages all the unique strengths of Java, such as platform independence, multithreading, interfaces, and introspection. It allows developers to combine rule-based and object-oriented programming to add business rules to new and existing applications. ILOG JRules for Java fully supports JDK v 1.2.2 through J2SDK v 1.3.1, Enterprise JavaBeans 1.1, JDBC 1.2, Servlets 2.2, J2SE, and J2EE.

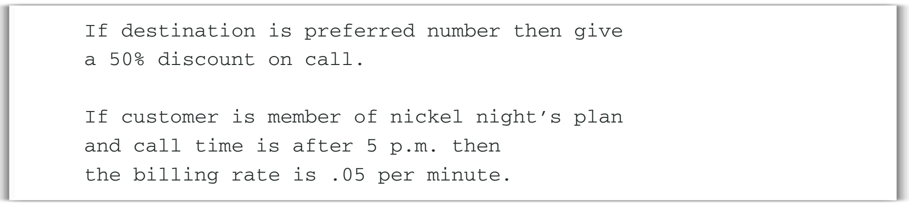
7.2.1 Writing Business Rules

To be effective and manageable, the rules need to be expressed in a language perfectly understood by the user. ILOG JRules provides a Business Rules language dedicated to business users and a Java-like language dedicated to Java developers.

7.2.1.1 Business Rule Language

When it comes to describing market conditions inserted in a business process, no business analyst wants to learn Java.

ILOG JRules is the only rule engine that provides a flexible and extensible business rule language — a rule language with a readable business level syntax. A business rule language uses a business rather than a technical vocabulary, and allows reasoning on an object model that reflects the structure of the business domain rather than the underlying Java implementation. This places the command of business rules in the hands of business users such as process analysts. The language can be made as simple or complex as necessary, providing business users with the freedom to create complex rules or constraining them to work within a predefined scope of the business domain. The combination of the rule editor and customizable business rule language creates one of the most powerful features available in today’s rule engines. Maintaining business rules is no longer quarantined to the realm of application developers, but open to the domain of business users themselves. Instead of typing in rules, they are constructed point and click, customized to the terms and vocabulary of the business world.



```
If destination is preferred number then give
a 50% discount on call.

If customer is member of nickel night's plan
and call time is after 5 p.m. then
the billing rate is .05 per minute.
```

Example of rules expressed in language meaningful to business users

7.2.1.2 JRules Rules Language

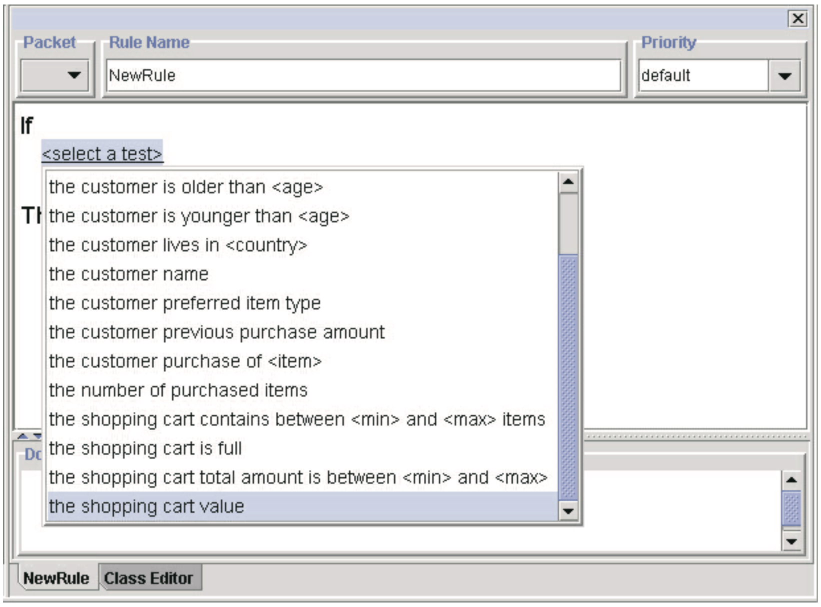
ILOG JRules uses a sophisticated language, the ILOG Rule Language, which features Java-like syntax, XML support, and powerful language extensions. There are forty keywords in the ILOG Rule Language. Combining these with the features of Java provides the flexibility to implement any business logic quickly and easily.

ILOG JRules uses Java as the basis for its rule language and exploits all of Java's object-oriented features, including common expressions and tests, interfaces, arrays, loops, and scope management. Literals have the same syntax as Java, and can be values that are Boolean, integer, floating point, character, and string.

When supplementing a workflow engine with a rule engine for adding flexibility in the resource assignment or for matching events during enactment, Java developers simply use this language to benefit from the complete power of expression that JRules offers.

7.2.1.3 Rule Builder

One of the main features of the Rule Kit is the Rule Builder. It is the main venue for creating business rule applications that include ILOG JRules, and is a fully integrated graphical environment for developing and debugging ILOG business rule engines. The Builder is highly customizable. The look-and-feel of the interface can be customized through property files read when the Builder is launched. For instance, the content of menus and toolbars or the colors and fonts used in the GUI can be modified. Graphical styles used in the rule editors, colors and fonts, highlighting, and graphics used in debugging, error, and breakpoint marking are also adjustable.



Packet

Rule Name: NewRule

Priority: default

If
<select a test>
Th
the customer is older than <age>
the customer is younger than <age>
the customer lives in <country>
the customer name
the customer preferred item type
the customer previous purchase amount
the customer purchase of <item>
the number of purchased items
the shopping cart contains between <min> and <max> items
the shopping cart is full
the shopping cart total amount is between <min> and <max>
the shopping cart value
Do
NewRule
Class Editor

Figure 19 – Point-and-click Business Rule editor for business users

7.2.1.4 From Process Definition to Rules Definition

By combining JViews and JRules, the business user can integrate a new category of activities called decision nodes in the process model. These nodes are implemented as applications running business rules that can modify data or decide on the process routing according to market-sensitive or complex conditions that would be difficult to represent with formal processes.

Through simple mouse clicks on decision nodes, the business user can access the Rule Builder to create, modify, or simply read a set of business rules.

Thus, during the definition phase, expert business users can smoothly define both processes and business rules with graphical tools.

7.2.2 Integrating a Rules Engine

After having seen how business rules can be written, we will now consider how a rule engine works and how it can be integrated in a Java application.

7.2.2.1 Seamless Java Integration

While traditional rule-based systems often require a proprietary language to define the objects used by the inference engine, the ILOG JRules engine directly infers from Java objects in the application – *without any duplication*. There is no restriction on the types of classes that are usable in the ILOG Rule Language.

The rules can therefore directly reference the Java objects handled by a workflow engine such as activities, resources, process instances, and so forth. The integration of rules in a workflow engine, for instance to add real-time resource assignment policies, becomes trivial.

Similarly, business rules inserted in decision nodes can transform data or route processes based on objects representing the order, the customer and his history, and the current promotion campaign. These objects can be implemented with Java or XML.

7.2.2.2 XML Binding

ILOG JRules is an XML-centric rule engine, meaning that an XML binding mechanism is available that allows rules to be written that reason directly on XML objects. This functionality has been integrated into the core of the rule engine, enabling the rule engine to treat XML objects just as it does Java objects. Since the rule engine parses the XML into its own representation, there is no need for adapters or translation code (such as DOM, SAX, or JDOM). XML objects can be directly referenced in rules with the same syntax as for Java objects and can be asserted, retracted, and updated, just like Java objects.

This feature is particularly important in the context of BPM, as many standardization organizations are turning to XML languages for process definition – like the BPML from BPMI.org, XPDL from WfMC, or ebXML from ebXML.org.

Business rules can then directly match and transform XML activities, actors, data, and transitions supported by these emerging standards.

7.2.2.3 EJB Integration

> ***As EJBs have taken hold, so has customer demand to lace these discrete business components together into end-to-end business processes. Calling an EJB is now a standard checklist item for the leading business process automation tool vendors.***
>
> ***Jane Stanhope, Giga Information Group***

ILOG JRules is EJB and J2EE ready, benefiting from all of the middleware services provided by the application server – load balancing, persistence, and fail over. ILOG JRules is the only rule engine that can be embedded directly into Enterprise JavaBeans, both entity and session beans. ILOG JRules provides the most flexible rule engine integration available for EJBs, including a rule engine server packaged as an Enterprise Bean.

7.2.3 Performance

7.2.3.1 Highly Optimized Rule Engine

ILOG JRules optimizes the Rete algorithm for the Java language, providing unsurpassed execution speed for large numbers of rules. The Rete algorithm is widely used because of its ability to handle large amounts of rules within an application and its unsurpassed performance in handling rules that reference dynamically changing data.

7.2.3.2 Compiled Rules

ILOG JRules is the only rule engine supporting compiled rules for applications that require high performance. Rules can be translated directly into Java classes and integrated into the application to improve performance of the rules by a factor ranging from 4 to more than 10.

8 Conclusion

Embedding workflow technology in BPM solutions is not enough to realize the promise of the Internet. The definition, supervision, and optimization of automated businesses must highly involve business managers – and not only technical people – so that their sense of business and their expertise is reflected in the implementation.

To achieve this goal, BPM tools must:

- Improve their user interfaces so that:
  - Business managers can efficiently participate in process definition with minimal training (intuitive formalism and icons, automatic diagram layout, simulation)
  - Business managers and participants can monitor the execution of processes and react to unexpected events
  - Business managers can improve their processes by analyzing measurements obtained during enactment
- Introduce business policies in a non-intrusive way so that:
  - Business managers can define rule-based activities that express business policies easily adaptable to market conditions
  - Complex transition conditions that reflect the real business complexity are easily expressed with rules
  - Bottlenecks are reduced with dynamic reallocation rules added to the enactment system

After more than 14 years of worldwide leadership in the object-oriented components market, ILOG proposes two excellent pure Java products – ILOG JViews and ILOG JRules – that enable software developers to rapidly develop BPM tools with the above benefits.

8.1 Features and Benefits Obtained by BPM Tools Through ILOG Components

8.1.1 Features

BPM vendors and integrators can expect to provide the following features to their business end-users in record time:

BPM Tool	*Features Obtained by using ILOG JViews and ILOG JRules*
Process Definition	• Intuitive tools used by all participants, with little training – True-to-life symbols – Editing aids that clearly depict relationships – Automatic layout of process diagrams – Expand/collapse subprocesses – Natural expression of transitional rules • Support enterprise look-and-feel • Connectivity to simulators with graphical animation • Rule-based activities for expressing business rules
Process Administration	• Display of process diagrams for documentation purposes • Easy access from browser clients
Workflow Enactment	• Dynamic and flexible assignment policies to reduce bottlenecks
Work List Handing	• Diagram displays and backlog indicators to help participants better understand their roles and degrees of urgency of work items • Display the current status and position of a specific item in the workflow • Easy access from browser clients
Business Activity Monitoring	• Tabular and chart views supplemented by diagram views that give all the relevant information needed to monitor a process instance or a set of instances • Animated symbols to identify activity status and bottlenecks of the system
Business Performance Analysis	• Executive Dashboard to report consolidated data through various graphical views and access detailed measurements • Statistical performance views of a process execution during time intervals • Reuse of process diagrams as designed during the definition phase to help detect the cause of problems • Charts and timelines represent process execution and resource usage over time

8.1.2 Benefits

The benefits expected by BPM tools supporting such features include:

- By providing a unified interface and support for process analysis, refinement, implementation, and monitoring, ILOG JViews for Workflow generates savings that can reach more than 50% of the design and refinement expenses, together with increased end-user participation, shorter analysis and design cycles, deeper analysis, and more frequent and adjusted continuous changes to match market demands.
- By smoothly integrating tools for defining business processes and tools for defining business rules, business experts are placed in the center of the business process automation life cycle. The resulting services not only match the business organization, but also the market environment and company policies. The clean separation between business logic and application code or business processes enable fast adaptation to market changes by business users, without service interruption or costly implementation.

8.2 Why ILOG?

Selecting ILOG JViews and ILOG JRules is a strategic advantage for workflow vendors. It enables them to build a BPM system of unprecedented ease and control that places the business user in the center of the process automation life cycle.

8.2.1 A Solid Technology

ILOG JViews and ILOG JRules provide high-level components that address advanced user interface and business rule needs, while giving low-level control of these components through a comprehensive API. As a result, developers save time by using ready-to-use components, but are still able to extend functionalities and satisfy the application-specific requirements.

Since some functionalities (such as graph layout) are too complex to implement efficiently, most diagramming tools simply omit this must-have feature. With ILOG JViews, no specialists are required in-house; ILOG's specialists have put their expertise into the product so that it can be embedded.

8.2.2 Saving Development Costs

Development and maintenance risks are also drastically reduced since JViews and JRules are used by thousands of developers and fully supported by ILOG professional services and R&D team.

This is illustrated by the following points:

- Only 2 weeks (1,300 lines of code) are necessary to develop a Workflow Modeler with ILOG JViews components.
- 2 man-months were enough for abaXX Technogogy AG, an e-business solution provider, to develop and integrate a user-friendly process modeler into its product line.
- 2 man-months were enough to develop a front-end for business process monitoring with animated symbols, dual views showing results as tables, diagrams, and charts.

8.2.3 Empowering Business Users

As stated in the previous sections, workflow technologies are spreading among more and more applications and businesses. The successful implementation of processes implies the active participation of business experts and participants who need appropriate definition and monitoring tools.

The market will appreciate BPM vendors who empower business experts with tools that help them:

- Define efficient processes
- Integrate business policies through a simple business rules language

- Monitor their execution
- Make decisions based on execution results to improve the business processes

8.2.4 From a Company you can Trust

The ILOG choice is a safe choice for ISVs and SIs who need long-term relationships with their partner suppliers:

- ILOG JViews and ILOG JRules are the result of ILOG's 14 years of experience in building object-oriented software components for visualization and business rules that thousands of customers worldwide use in mission-critical applications. In fact, ILOG produced ILOG JTGO, an ILOG JViews specialized component for telecommunications network management displays that is now the de facto standard for operators and manufacturers.
- ILOG customer support centers and consultants are available worldwide to satisfy customer demands and solve their problems.

The development of ILOG JViews and ILOG JRules in the Workflow and BPM market is making available to the user community best-of-breed front ends and business rules that always increase ease of use and involvement of business experts.

> ***"The ILOG approach has every chance of succeeding, and will provide a fundamental shift in the growth rate of the workflow market by making workflow available to an unlimited number of applications in all the economic sectors."***
>
> ***Martin Ader***

ILOG Worldwide Information Center - Tel: +1 775 831 7744 or +1 800 367 4564 • URL: http://www.ilog.com

France ILOG S.A. - Gentilly - Tel: +33 (0)1 49 08 35 00 - E-mail: info@ilog.fr – **Germany** - ILOG Deutschland GmbH - Bad Homburg v.d.H. - Tel.: +49 6172 40 60 - 0 - E-mail: info@ilog.de – **Japan** - ILOG CO., LTD - Tokyo - Tel: +81 3 5211 5770 - E-mail: info@ilog.co.jp – **Singapore** - ILOG (S) Pte Ltd - Singapore - Tel: +65 773 06 26 - E-mail: info@ilog.com.sg – **Spain** - ILOG SA - Madrid - Tel: +34 91 710 2480 - E-mail: info@ilog.es – **UK** - ILOG Ltd - Bracknell - Tel: +44 (0) 1344 66 16 00 - E-mail: info@ilog.co.uk – **USA** - ILOG, Inc. - Mountain View, CA - Tel: +1 650 567-8000 - E-mail: info@ilog.com

Representatives and distributors in other countries

ILOG™

11/01