Slides contain material from www.yawlfoundation.org
Developed by Wil van der Aalst, Michael Adams, Arthur ter Hofstede, Nick Russel

WORKFLOW MANAGEMENT: YAWL OVERVIEW AND ARCHITECTURE

Business 2710 – Class 13

Learning Objectives

- Be able to describe the architecture and capabilities of YAWL
- Be able to characterize the YAWL system in generic WfMS terms

Textbook Chapter 1

Textbook Chapter 7

Textbook Chapter 23

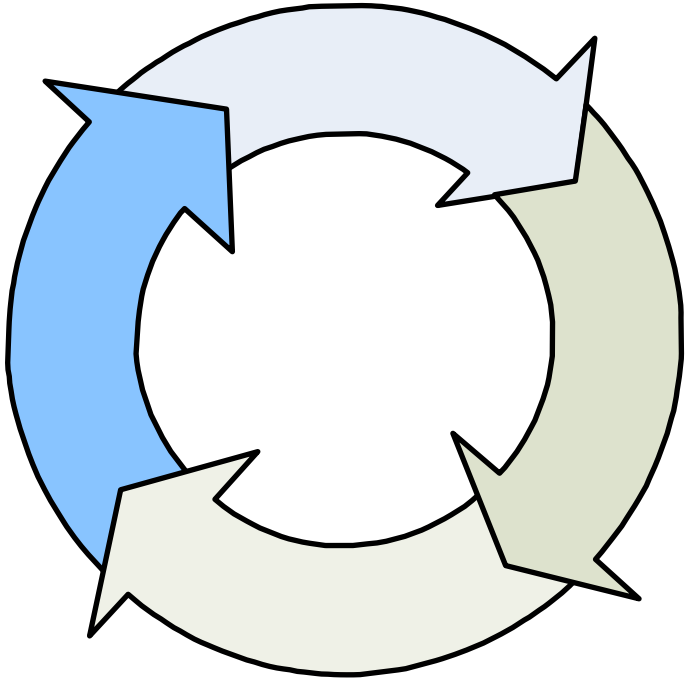
Business Process Automation

- Compliance with process model
- Explicit representation of control-flow
  - Easier process changes
- Explicit representation of resource involvement
  - Routing of work to right resources
  - Workload and work history can be taken into account in work assignment
- Monitoring support
  - Identification and resolution of bottlenecks
- Post-execution analysis (process mining)
  - Identification of opportunities for process improvement

Process Automation Life-Cycle

diagnosis

process enactment and monitoring



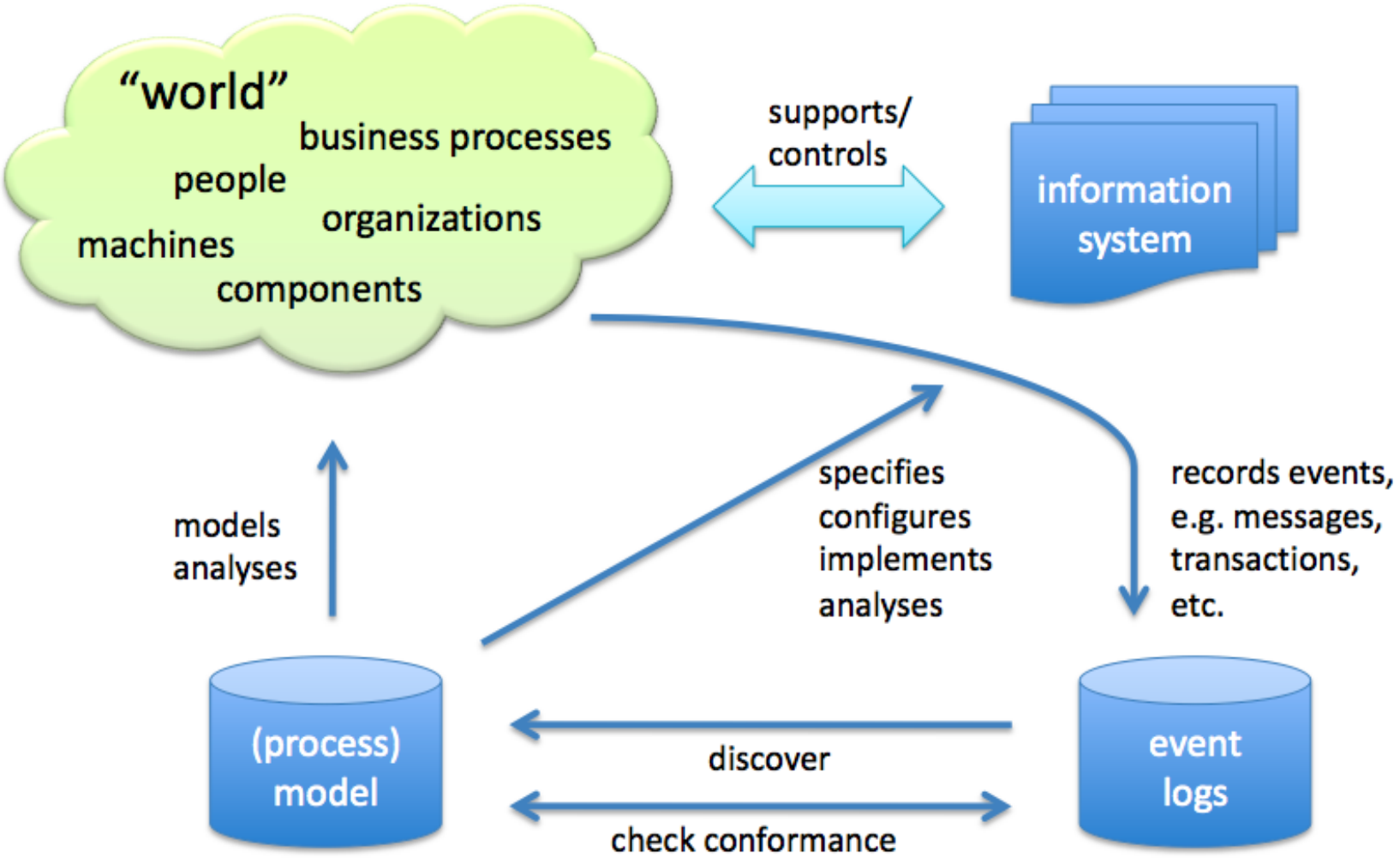
process (re)design and analysis

system configuration

Process Models

- Insight and understanding
- Analysis
  - Performance
  - Correctness
- Execution

Role of Process Models



Business Process Management

- Many approaches for describing process
  - Not all support automation
- Process complexity
  - Process modeling languages lack necessary concepts
- Lack of standardization
  - Incompatible and inconsistent interpretation
- YAWL as an independent, complete language

BPM Standards

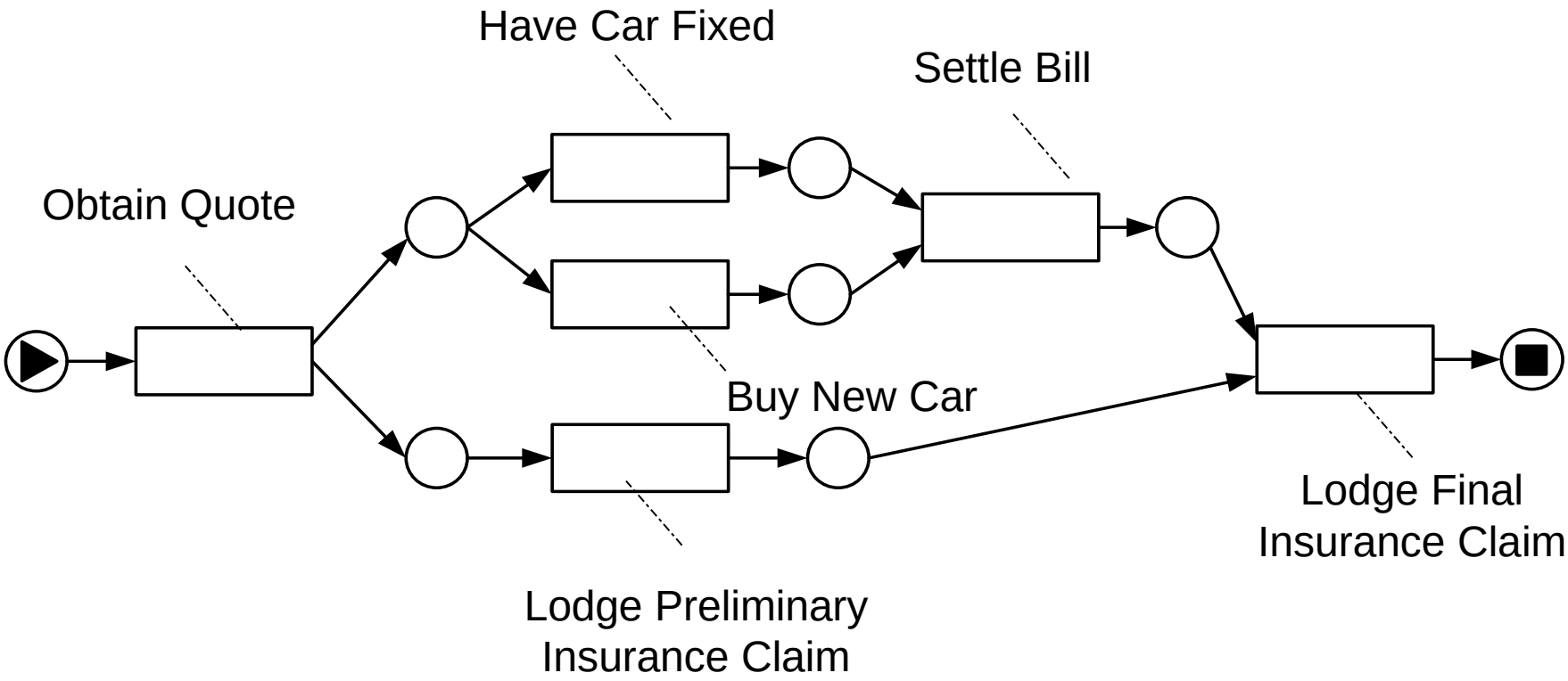
- XPDL
- BPEL
- BPMN
- EPC
- UML

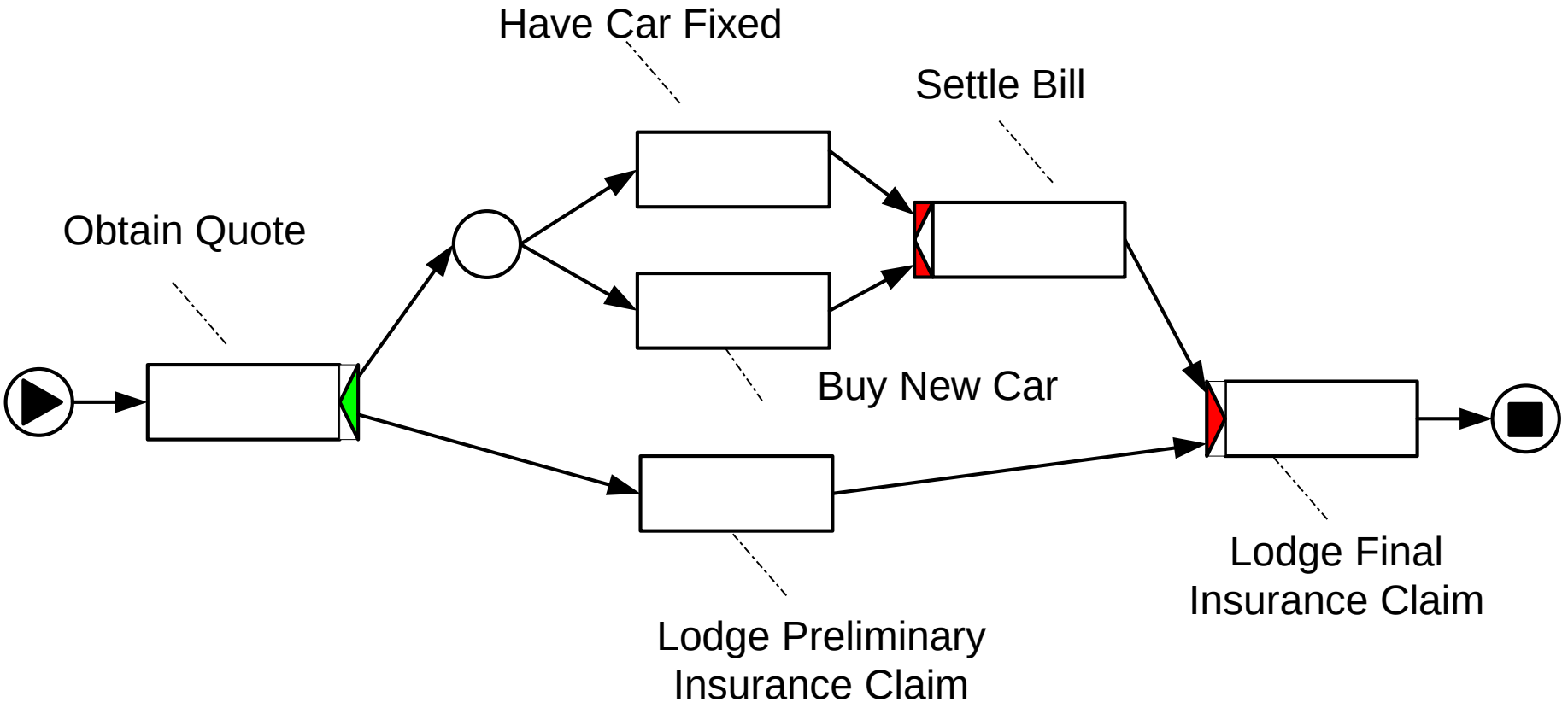
Workflow Pattern Initiative

- Standard language
- Based on existing workflow systems
- Identifies necessary elements
- Evaluation and comparison of workflow modeling language and workflow systems

YAWL

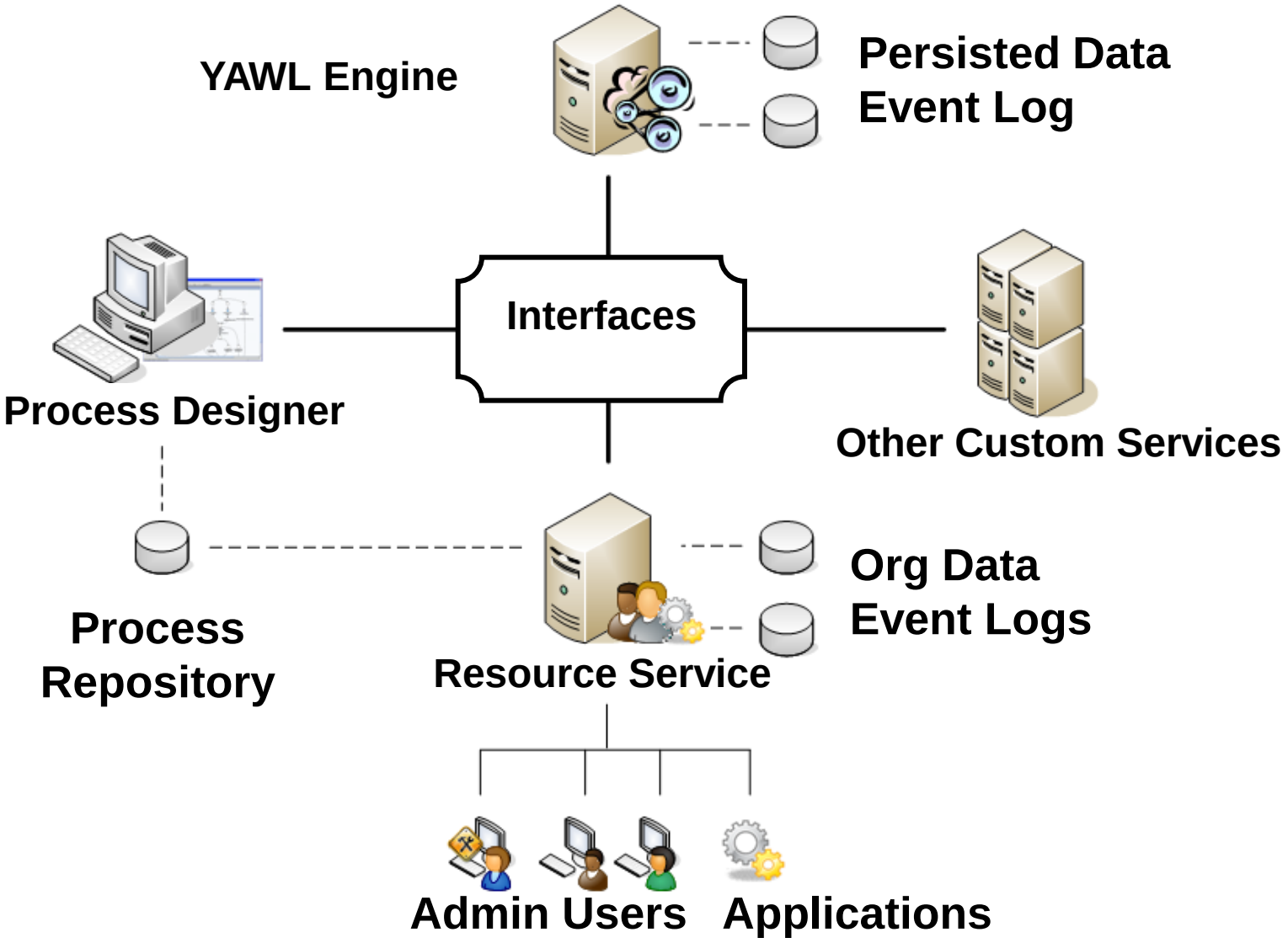
- Extension of Petri Nets and Workflow nets
- Cancellations (“Reset nets”)
- Multiple instances
- Synchronization of active branches (“OR”)
- Formal and precise semantics

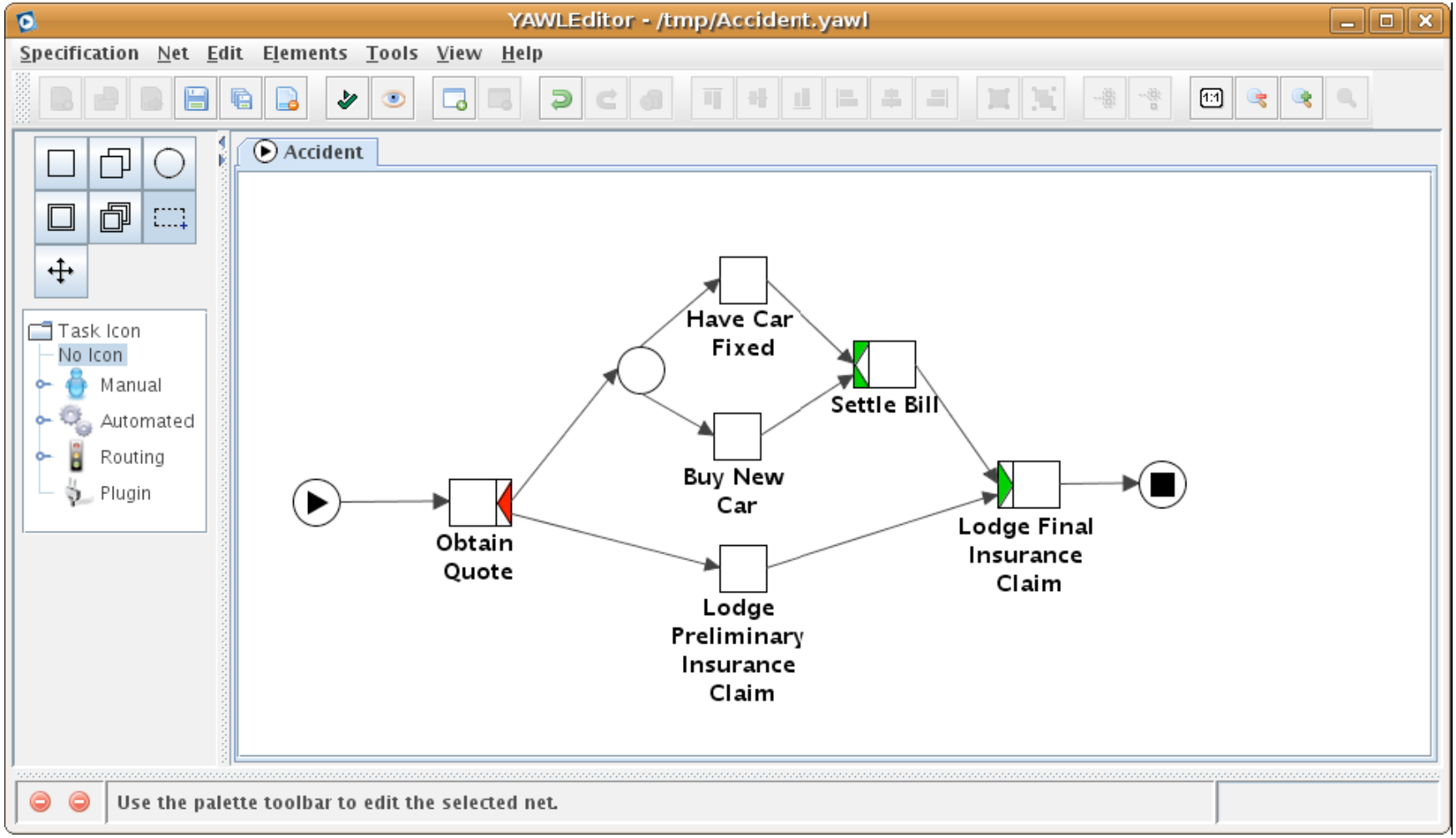

Have Car Fixed
Settle Bill
Obtain Quote
Buy New Car
Lodge Final
Insurance Claim
Lodge Preliminary
Insurance Claim

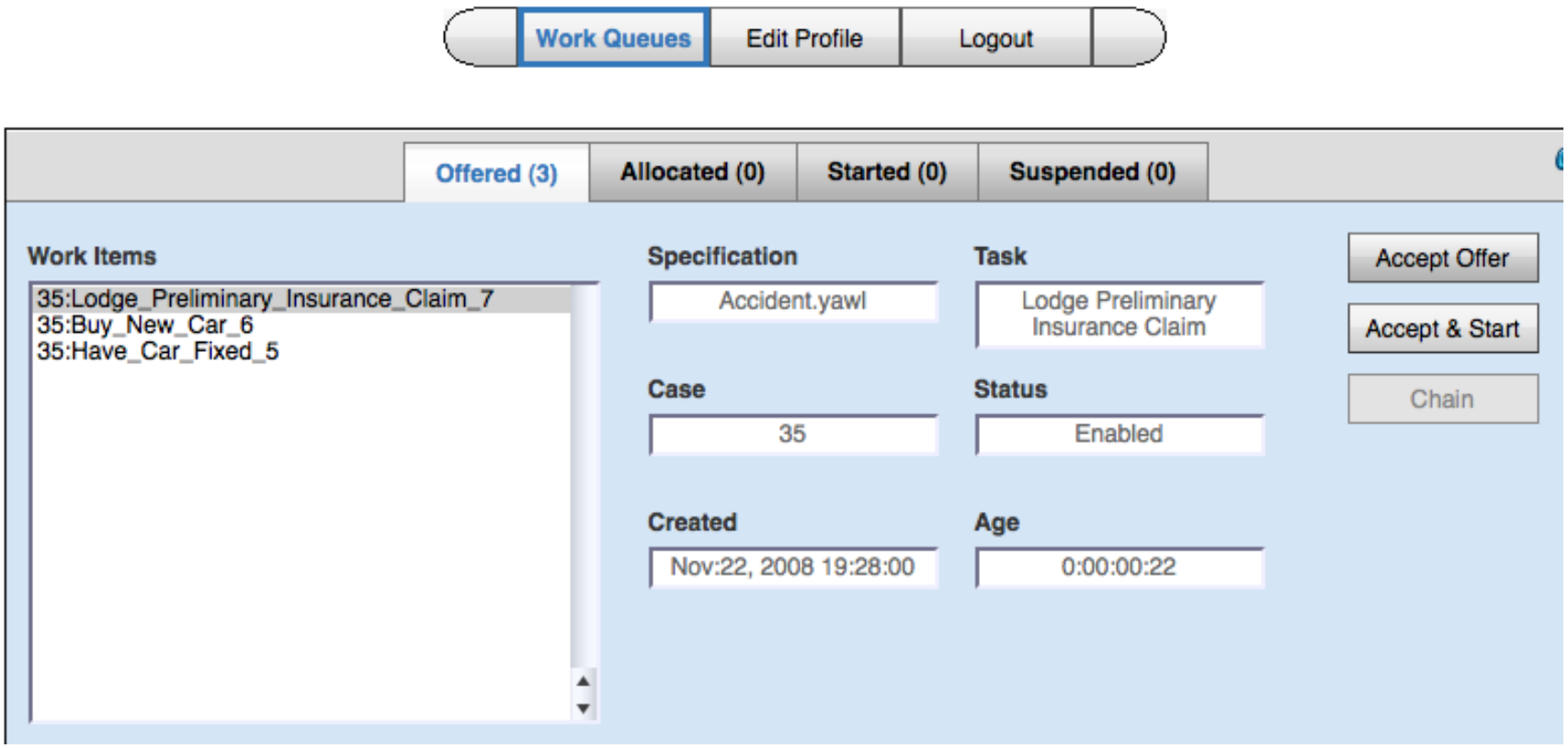

Have Car Fixed
Settle Bill
Obtain Quote
Buy New Car
Lodge Final
Insurance Claim
Lodge Preliminary
Insurance Claim

YAWL System

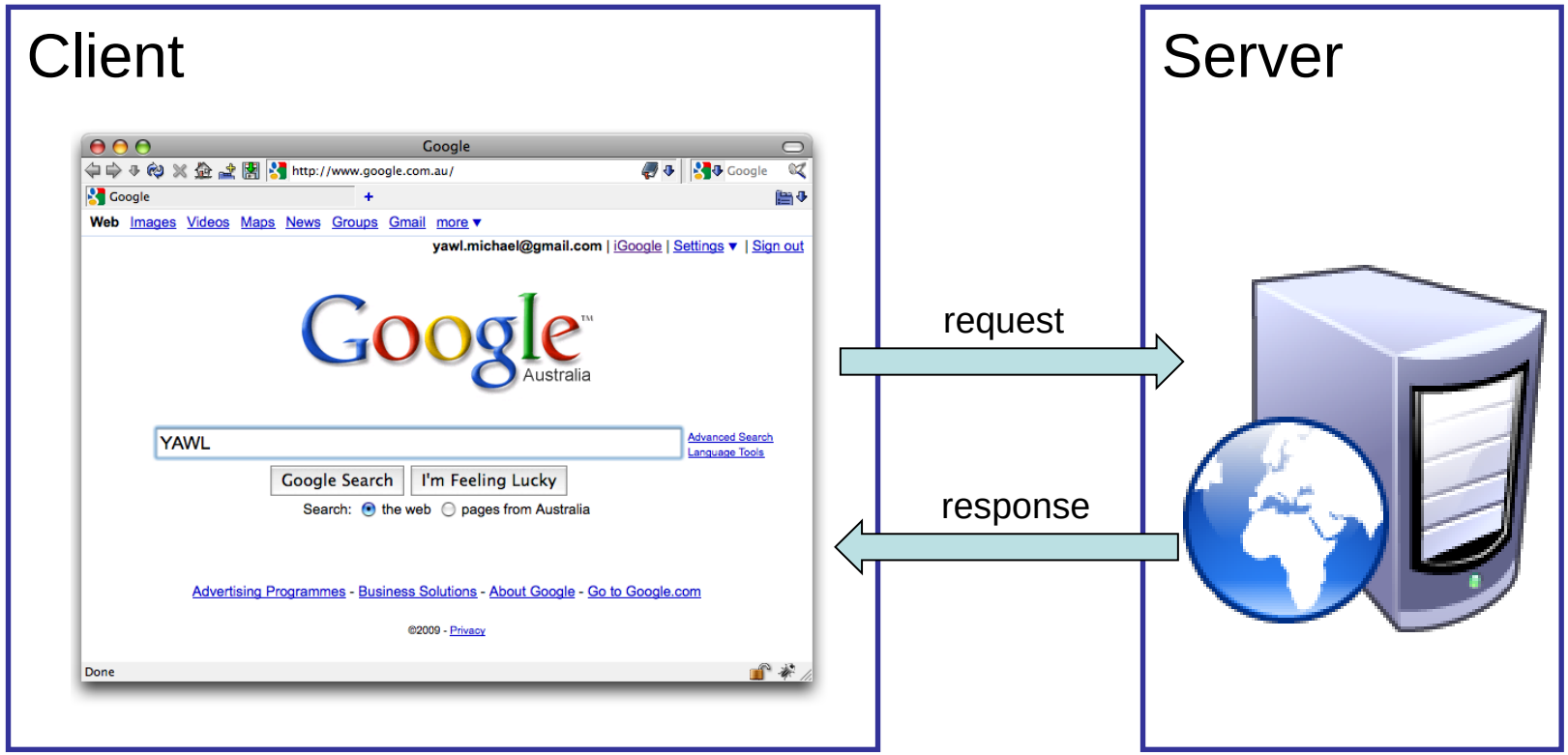
- Build Time
  - Editor
- Run Time
  - Execution engine (control flow and data)
  - Resource service (resources)
  - External services (codelets, web-services)
- Open source, multi-platform
- Flexible (exceptions, declarative workflows,…)

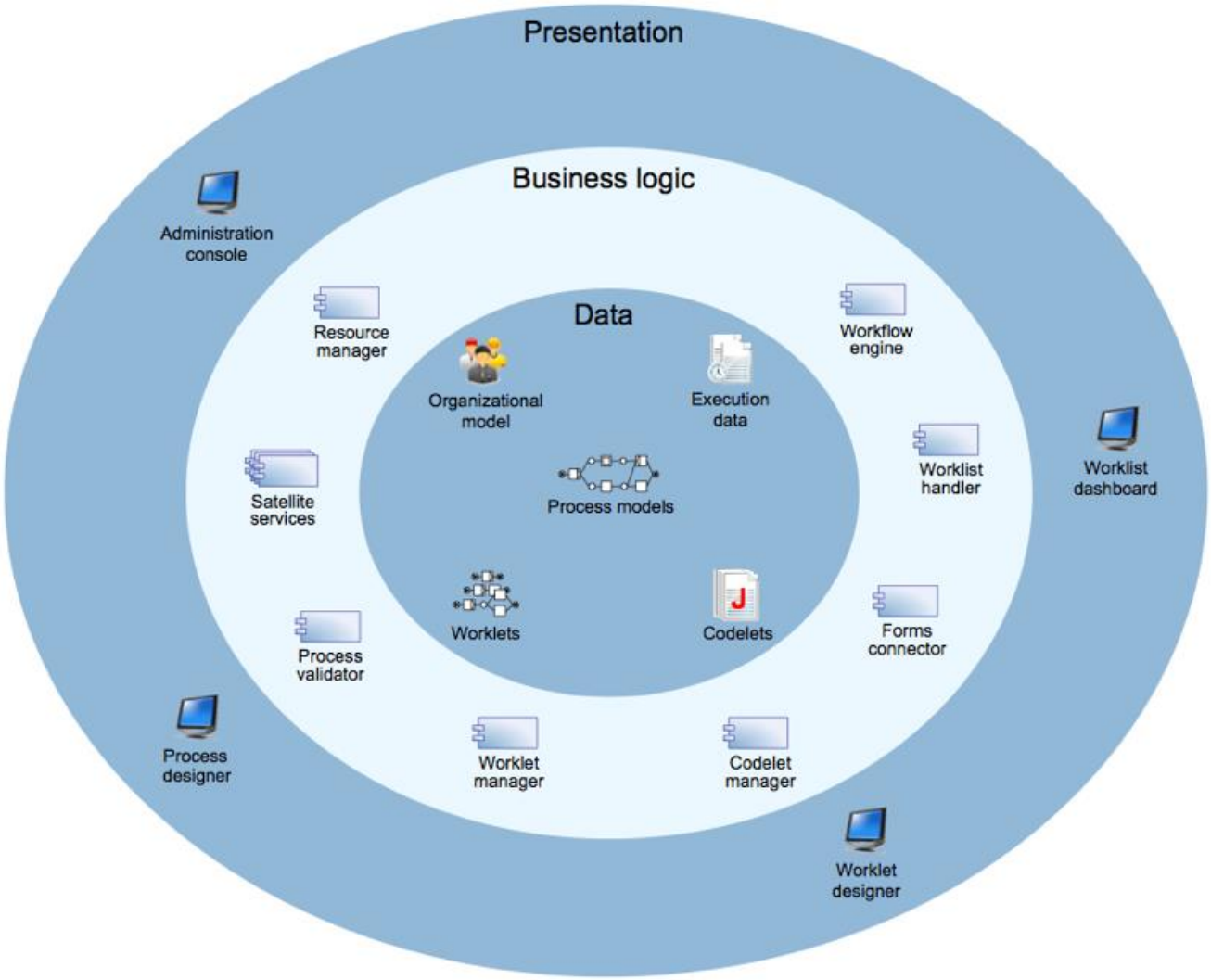

YAWL Engine
Persisted Data
Event Log
Interfaces
Process Designer
Other Custom Services
Process
Repository
Org Data
Event Logs
Resource Service
Admin Users
Applications


YAWLEditor - /tmp/Accident.yawl
Specification
Net
Edit
Elements
Tools
View
Help
Accident
Task Icon
No Icon
Manual
Automated
Routing
Plugin
Have Car Fixed
Settle Bill
Buy New Car
Obtain Quote
Lodge Final Insurance Claim
Lodge Preliminary Insurance Claim
Use the palette toolbar to edit the selected net.

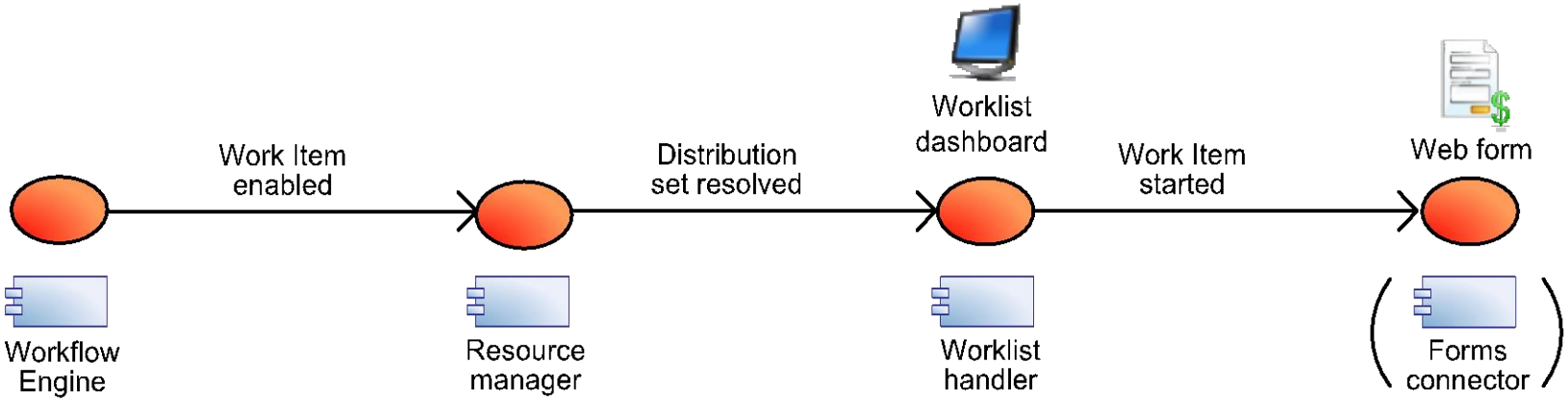

Work Queues
Edit Profile
Logout
Offered (3)
Allocated (0)
Started (0)
Suspended (0)
Work Items
35:Lodge_Preliminary_Insurance_Claim_7
35:Buy_New_Car_6
35:Have_Car_Fixed_5
Specification
Accident.yawl
Task
Lodge Preliminary Insurance Claim
Case
35
Status
Enabled
Created
Nov:22, 2008 19:28:00
Age
0:00:00:22
Accept Offer
Accept & Start
Chain

YAWL Architecture




Presentation
Business logic
Data
Administration console
Resource manager
Workflow engine
Organizational model
Execution data
Worklist handler
Worklist dashboard
Satellite services
Process models
Worklets
Codelets
Forms connector
Process validator
Process designer
Worklet manager
Codelet manager
Worklet designer

Typical Work Item Execution Path



The YAWL Engine

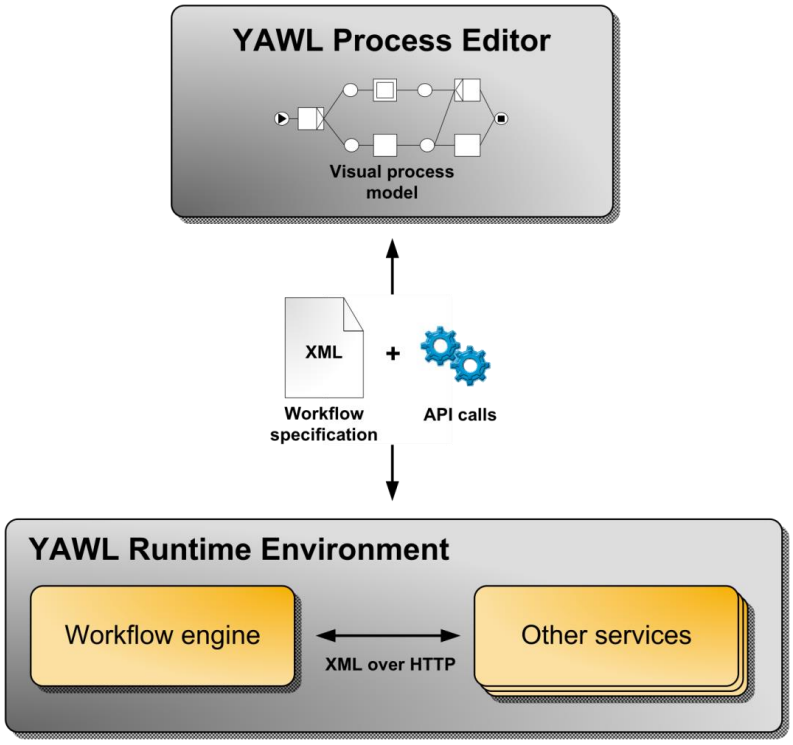
- The Engine manages the execution of instances (cases)
  - Each case is progressed according to its current state and control-flow description
  - Performs the specified data mappings between the case and its tasks as required
- At each stage of a process, the Engine determines which work items should be offered, and which events should be announced to the environment
  - Each task in a process instance is associated at design time with a YAWL Service (either explicitly or, if not specified, is implicitly associated with the default worklist handler).

The Yawl Editor

- The YAWL Editor is a Java desktop application for the creation and verification of YAWL process specifications.
- It communicates with a running Engine through *Interface A*, to obtain a list of the YAWL Services currently registered with the Engine.
- It also communicates with a running Resource Service through *Interface R*, to obtain lists of the various organizational resources and codelets currently available, so that selected resources can be associated with particular tasks.

The YAWL Editor

- The Editor provides a tool palette from which modeling elements (such as tasks or conditions) may be chosen.
- At any time, a process may be verified and analyzed to ensure completeness, soundess and so on.
- When complete, the process definition may be saved to a disk file, in XML format



The Resource Service

- Provides the resource perspective for specifications
  - Completely separate from the engine
- Basic role is to allocate work items to resources for processing
- It has four primary components:
  - **Resource Manager:** handles all the resource patterns
  - **Worklist:** a web-based user interface
  - **Forms Connector:** show either custom designed or dynamically generated forms for work items
  - **Codelet Server:** for executing codelets