

Building an Extensible Workflow System for Healthcare



WIHIR Research Seminar

William Malyk

December 13, 2006

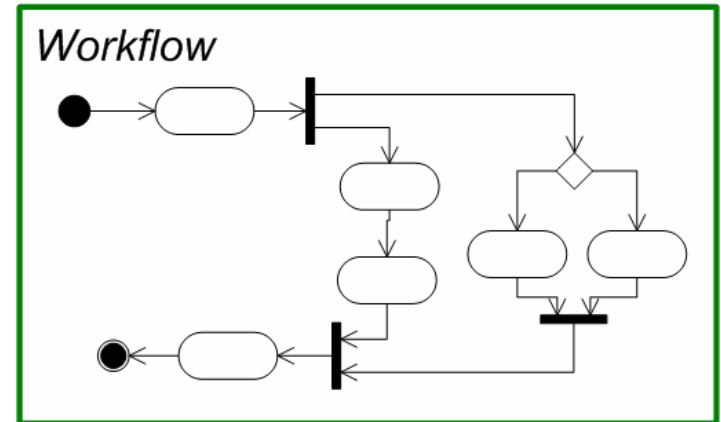
Introduction & Motivation

What is Workflow?

Workflow: the composition of individual tasks into larger processes.

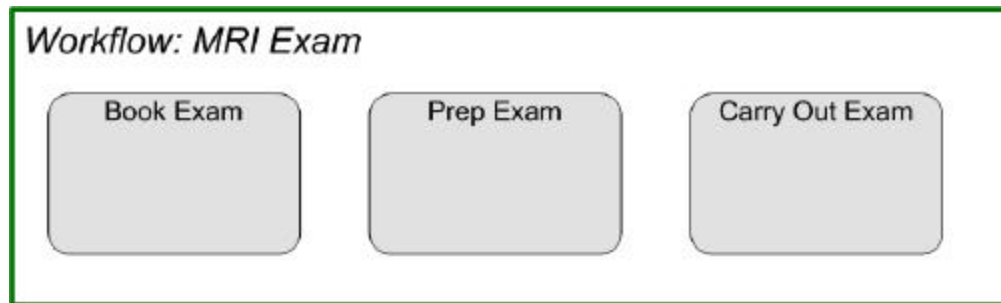
Examples:

- Processing a University application
- Having your car serviced
- Performing open-heart surgery



Workflow is Decomposition

- To make complex functions manageable workflow is decomposed into tasks or activities.

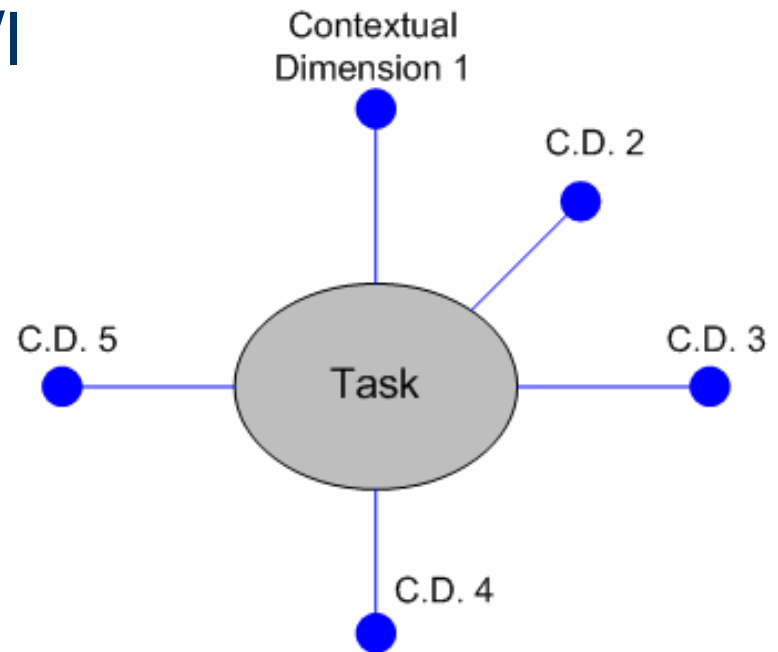


- **Workflow** is a systematic treatment of the **tasks** (units of work) which compose a process.
- Tasks are often specified in terms of:
 - content (what is carried out)
 - constraints
 - ownership (who carried/can carry out the task).

Workflow Context

- Contextual (Concerns/

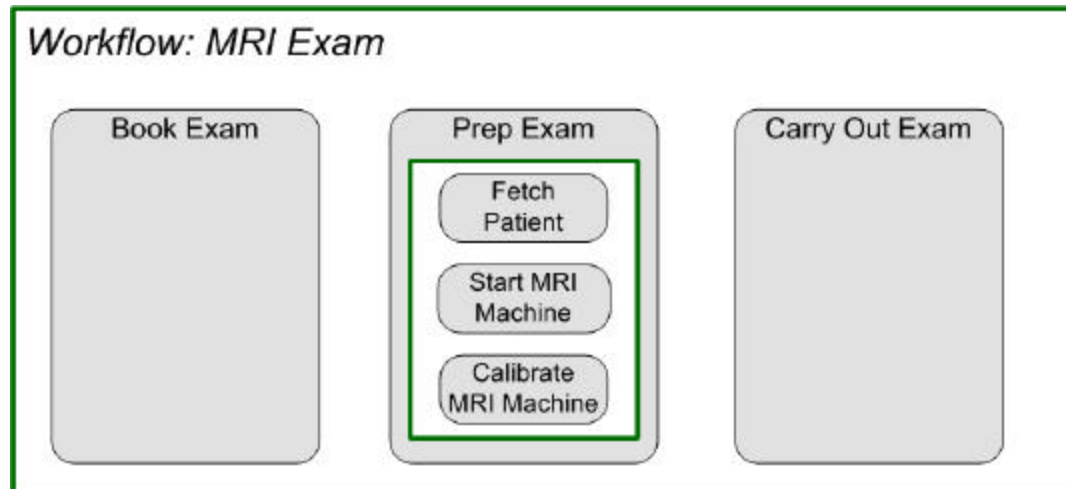
- Who
- What?
- When?
- Why?
- Where?



- Context gives meaning to tasks.

Workflow is Granular

- Workflow exists in levels.
- Considering workflow as a decomposition of one task into several smaller tasks immediately leads to a recursive relationship.



What is a Workflow System?

Workflow System:

manages workflows / processes, often allowing a user to reengineer their own processes.

Effectiveness of Workflow Systems

Current workflow systems work well... in some domains.

Examples:

- *Highly automated processes (BPM)*
- *Web-based workflow (SOA & WS)*
- *Banking and Manufacturing*

What is the Problem with Workflow Systems for Health?

Workflows in Healthcare exhibit unique characteristics

- Physicians are 'constrained by **guidelines**' rather than 'driven by rules'.
- Non-compliance with guidelines is widespread.
- Role relationships are intricate.

[QUAGLINI-2001]

Health is a complex and chaotic domain where traditional workflow practices do not consistently provide a solution.

The Bottom Line - Two Types of Workflow

- Workflow in chaotic environments, like those found in health, differs from 'traditional' workflow - therefore different representations and systems are required.
- In particular, we find there are two 'flavors' of workflow:
 1. Prescribed:
 - 'Traditional' Workflow
 - Workflow can be 'routed'
 2. **Assembled:**
 - Found in chaotic environments
 - Workflow is essentially 'history'
 - A deep representation context needs to be supported so that it, rather than routing, can drive the selection of tasks in a process

My Goal

Goal: to deliver a workflow system better tailored to the specific needs of the healthcare domain.

- To do this the system must:
 - Support the concept of assembled workflow
 - Support a rich notion of context/semantics (beyond simple routing) to drive workflow functionality
 - Support 'classic' workflow concepts to maintain interoperability

Outline for this Presentation

1. Overview of the Method
2. Outline the System Requirements
3. Detail the WoRK System Developed*
4. Discuss Open Problems & Future Work

A (Brief) Overview of the Method

Method Overview

The method for providing an improved workflow system for health was straightforward:

1. Develop Requirements
2. Design the System
3. Implement the System

Requirements of a Workflow System for Healthcare

Requirements

- Developed through a literature review
- In the interests of time only the key points will be elaborated on here (full details are available in Malyk-06).
- We will motivate the requirements by showing an example workflow management system (WfMS)

An Example WF System for Health

- Patient Workflow Management Systems (WfMS)
 - Patient outcomes are determined by:
 - The skill of the care provider
 - Collaboration inside the organization
 - WfMSs require:
 - Guidelines (GLs): "systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific circumstances" [FIELD, 1992]
 - An Ontology of the organization
 - To maximize outcomes the system tailors the GLs to the organizational needs.

WfMSs

- Patient Workflow Management Systems (WfMS)
 - Provide decision support
 - The nature of Healthcare generates unique requirements, the systems which fit the bill are sometimes termed careflow systems [QUAGLINI, 2001].
 - Role relationships are very complex in this setting
 - Non-compliance with GLs is widespread
 - Often workflows seem only to **exist after the fact** as histories/logs of what took place.
 - Workflow Mining can be used to recover a representation of this 'after-the-fact' workflow.

WfMSs

- Patient Workflow Management Systems (WfMS)
 - The GLs (Best Practices) are first put into a computerized representation
 - Usually the workflows are specified using a combination of visual techniques and Workflow Languages
 - The representation model is typically a Petri Net
 - The process of specifying a workflow is no trivial task:
 - It is a complex and lengthy process [AGRAWAL, 1998]
 - It is often flawed [AGRAWAL, 1998]
 - Workflow Mining can be used to simplify the specification task.

Example WfMS

- Dazzi et al created a Patient WfMS based on GLs in Italy in 97 [DAZZI, 1997]:
 - Produced a three tool system:
 - 1 - Editing Tool: support the specification of GLs
 - 2 - Translating Tool: from GL to a Wf Model (PN)
 - 3 - Low Level Wf Builder: ‘customize’ the PN to produce the specific Wf for the organization
 - Provided separation of concern between organizational expert and clinical process experts.

Dazzi WfMS

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

Example Editing Tool Interface [Figure 1 - DAZZI, 1997]

Dazzi WfMS

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

Example Petri Net Representation [Figure 2 - DAZZI, 1997]

Dazzi WfMS

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

Example PN for Specific Workflow[Figure 4 - DAZZI,1997]

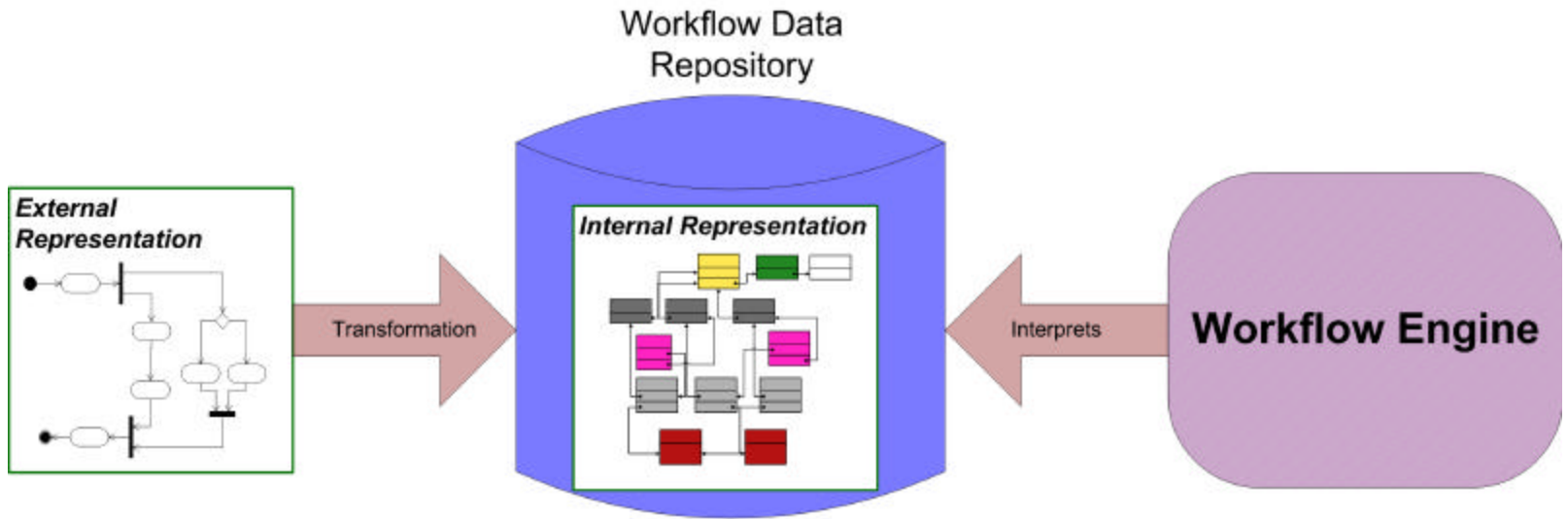
Modern Workflow Systems

According to Lawrence WF Systems come in three types:

1. External (separate app)
2. Embedded (acts as a monitor)
3. **Integrated (build around processes)**
 - Examples include the Jackson and Lawrence methods.
 - Feature two pieces:
 1. **Workflow Model / Representation**
 2. **Workflow Engine**

Traditional Treatment

- 1. **Composition stage** – External WF is authored.
- 2. **Translation stage** – External WF Translated into Internal WF.
- 3. **Enactment stage** – Internal WF is interpreted by the WFEngine.



1- Composition

2 - Translation

3 - Enactment

Question Becomes...

- Two fold:
 - What are the requirements of the workflow representation?

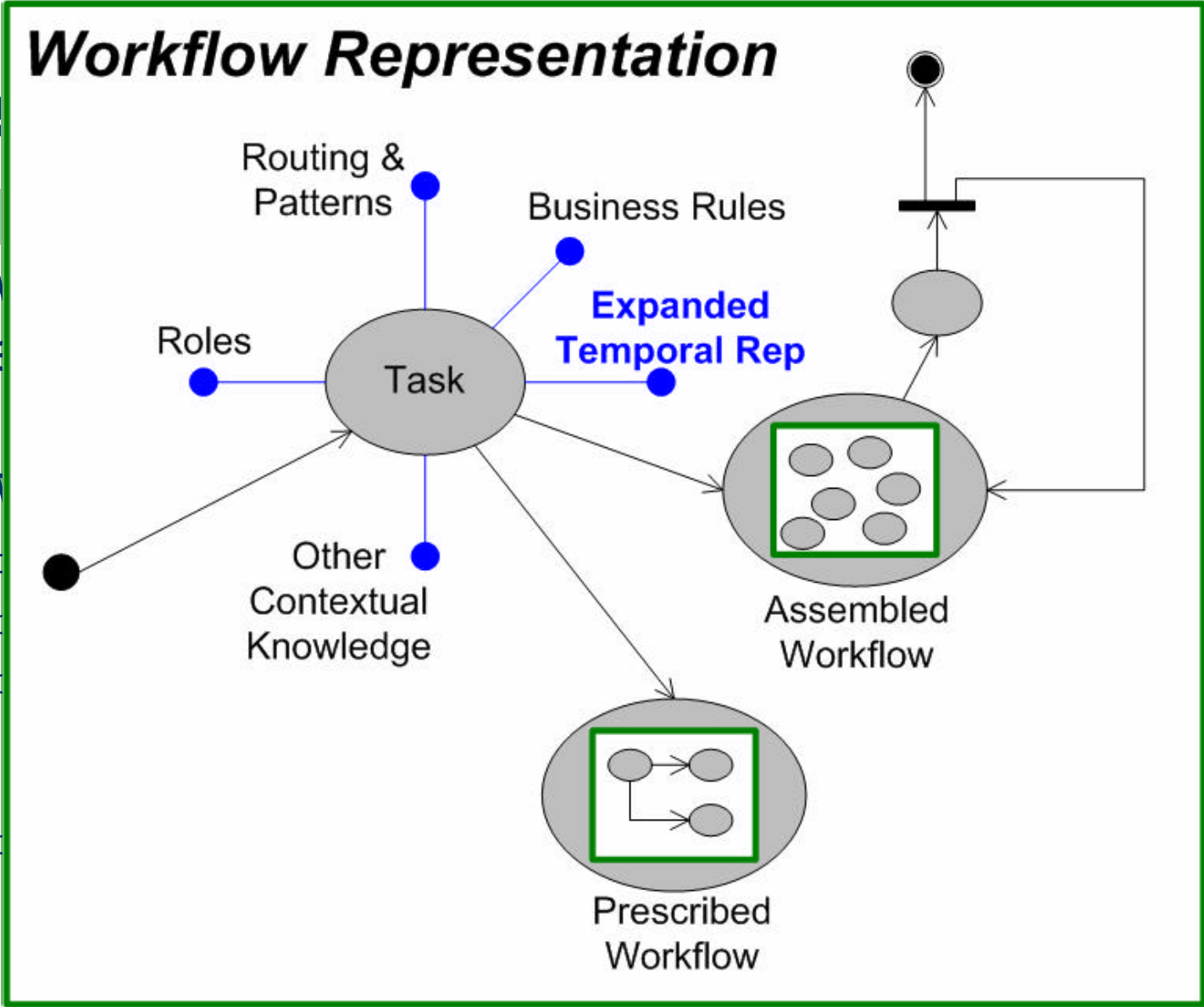
AND

- What are the requirements of the system which interprets the representation?

Re

- W
- re

- W



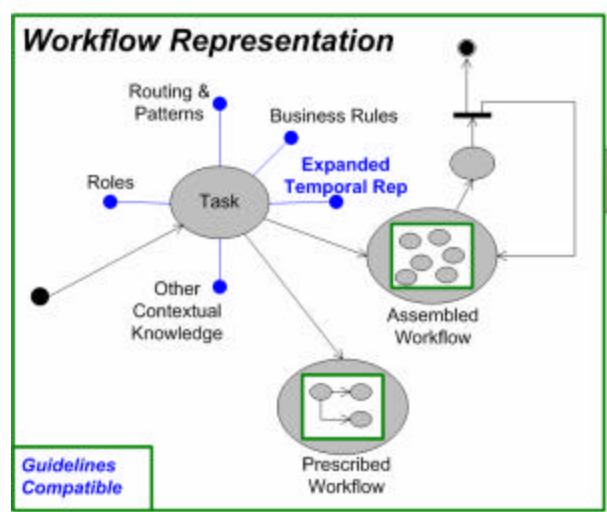
the

Requirements of Any WFE

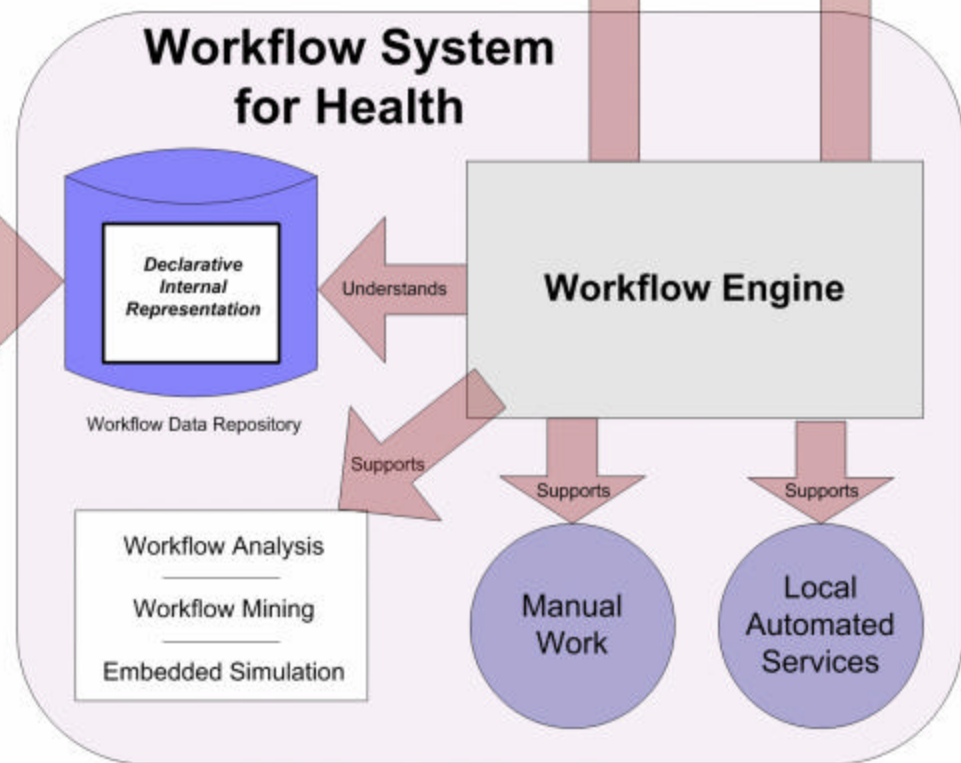
- Primary function: handle the execution of tasks.
- The engine must support the execution of:
 - Automatic Tasks - performed by software components.
 - Must decide when to execute the task.
 - Must activate the software component, pass it appropriate input data, monitor its execution and capture its results.
 - Manual Tasks - performed by people.
 - Must make appropriate tasks available to workers.
 - Provide them with required resources and capture result of the work.
- The most complex job performed by the WFE is deciding when/what tasks to execute.

Th

Workflow System for Health



Transformable



Understands

Supports

Supports

Supports

Supports

Supports

System Design

An Extensible Workflow System for Healthcare

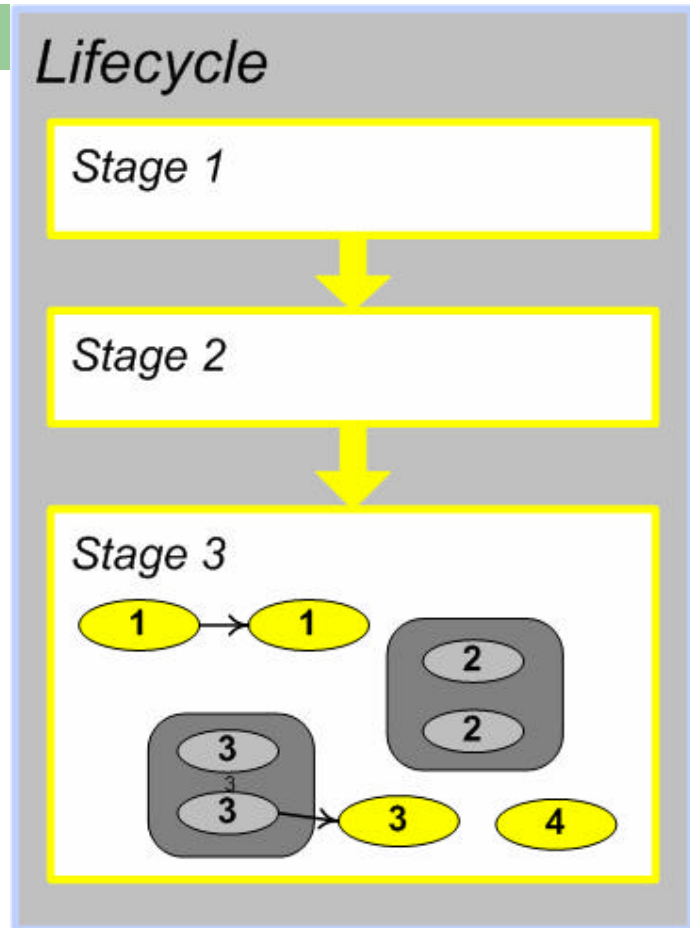
An Integrated Workflow System, composed of:

1. **A Workflow Model**
 - Based on the Jackson L-S-T model
2. **An Integrated Workflow Engine**
 - Programmed using a custom Domain Specific Language (DSL) for behavioral composition

Implemented as part of the Web Informatics Development Environment (WIDE) of the Computer Systems Group.

Workflow Model / Representation

- Consists of Entities, Lifecycles, Stages and Tasks
- These elements have a visual representation
- This representation can be represented directly in a database form.



Why this Workflow Model?

- De-emphasis on routing.
- Built to model reality.
- Can be implemented as data:

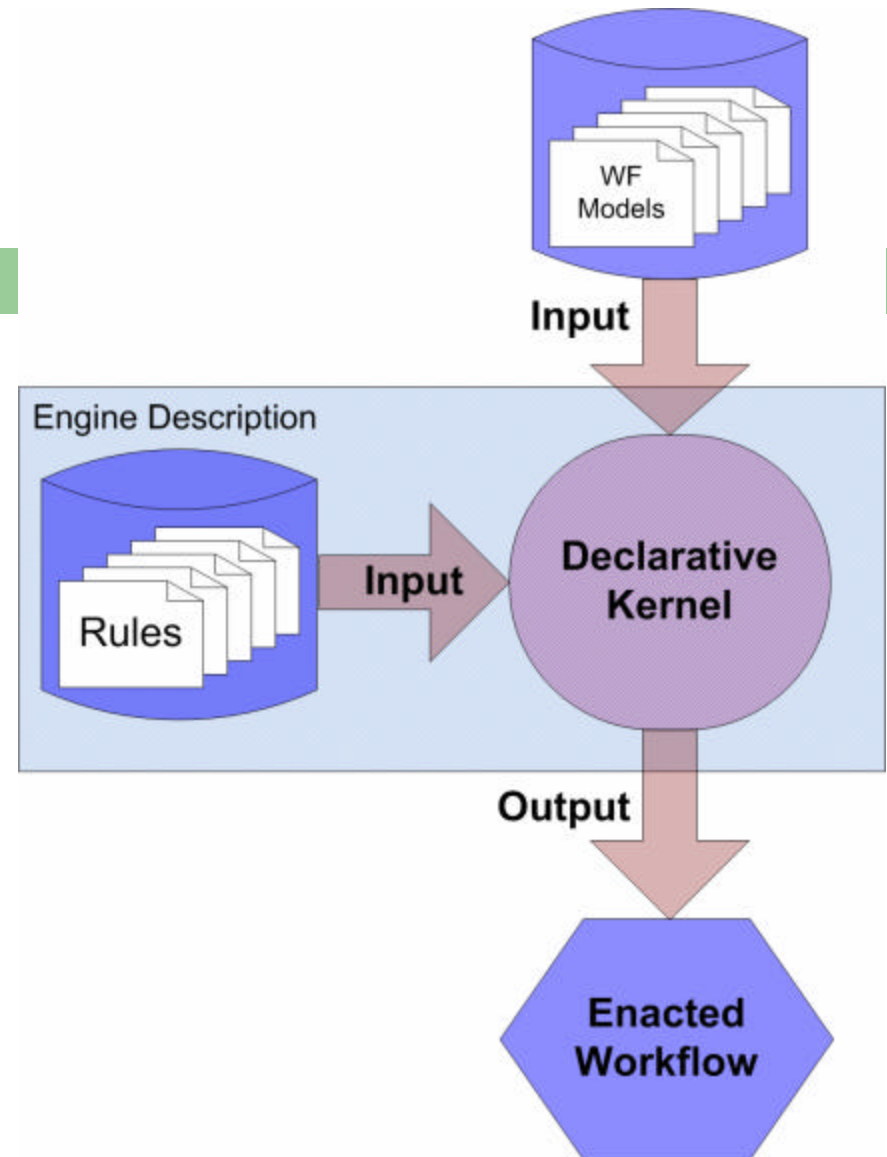
“Data is easier to change than programs.”

-Jackson, 1997

The Workflow Engine

Declarative Workflow Engine:

- Rules are specified in terms of behaviors.
- **The engine behavior itself is just data** ==> the engine can perform differently in different circumstances.



The Result

- A two-tiered DSL for workflow:
 - **Tier 1:** A language for describing workflow in terms of: Entities, Lifecycles, Stages, Tasks, Services, and Aspects*.
 - **Tier 2:** A high-level language for specifying the behavior of the workflow engine.
- An engine capable of transforming this language into a working system.

A sandbox for experimenting with workflows for the health domain.

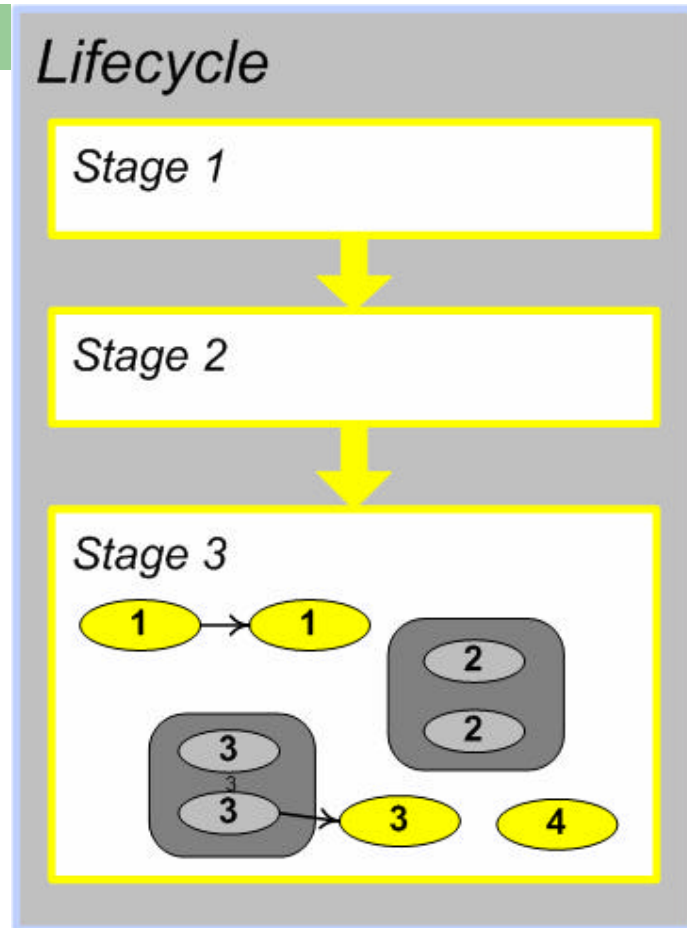
The Workflow System

The WoRK System

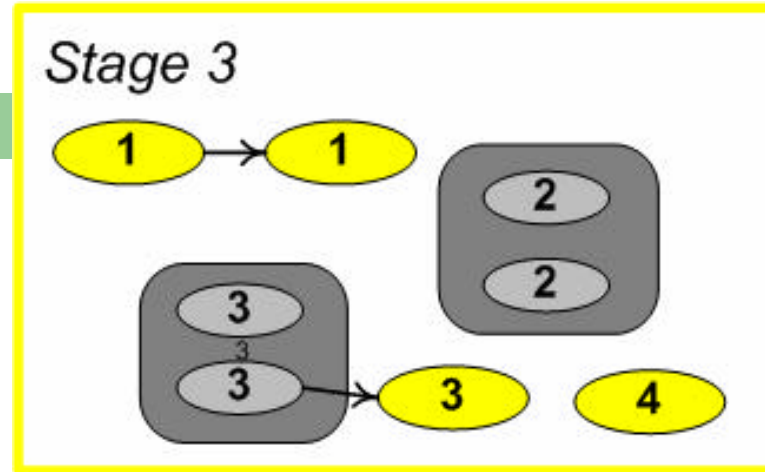
- Named after the Workflow Research Knowledge (WoRK) Group.
- Consists of:
 - A workflow model
 - A declarative workflow engine

Lifecycle-Stage-Task Representation of Workflow

- Every **Entity** in the system has one or more associated **Lifecycles**.
- Lifecycles pass through sequenced **Stages**.
- Stages consist of **Tasks** which are performed in parallel where possible.



Within a Stage: Task Sequencing / Routing

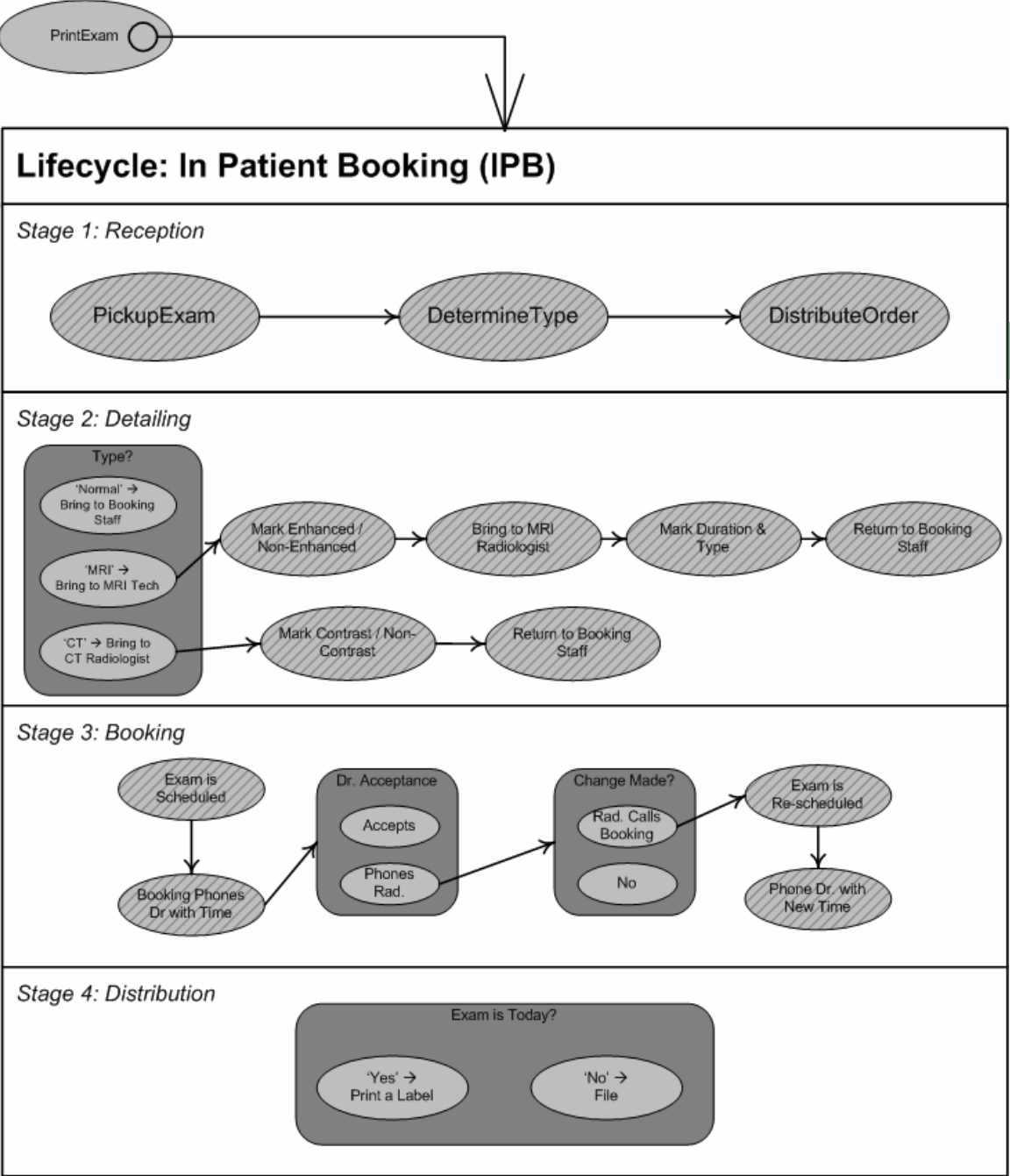


- Tasks become available (to execute) as soon as possible
- Tasks can have dependencies
- Tasks can spawn other tasks
- Tasks can be part of mutually exclusive groups

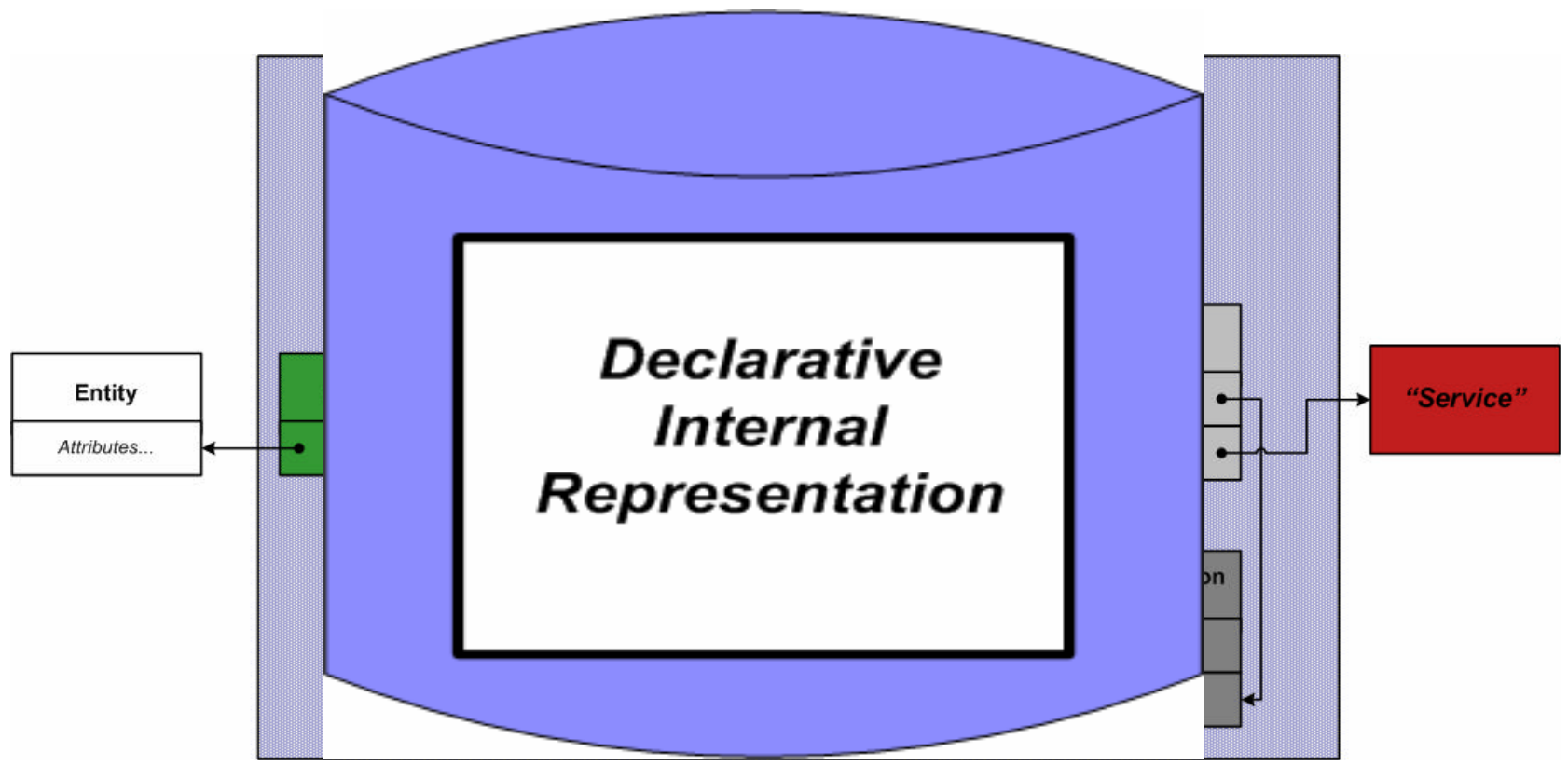
Deeper Considerations

- Tasks can be:
 - **Local Tasks**: automatic, manual
 - **Remote Tasks**: Web Services
 - **Complex Tasks** – have lifecycles of their own → Granularity of Workflow
- Workflows come in two flavors:
 - **Prescribed** – follow routing rules, have some type of worth in prescribing.
 - **Assembled** – workflows like those found in some areas of health where the emphasis is on capture of the process; where workflow is history.

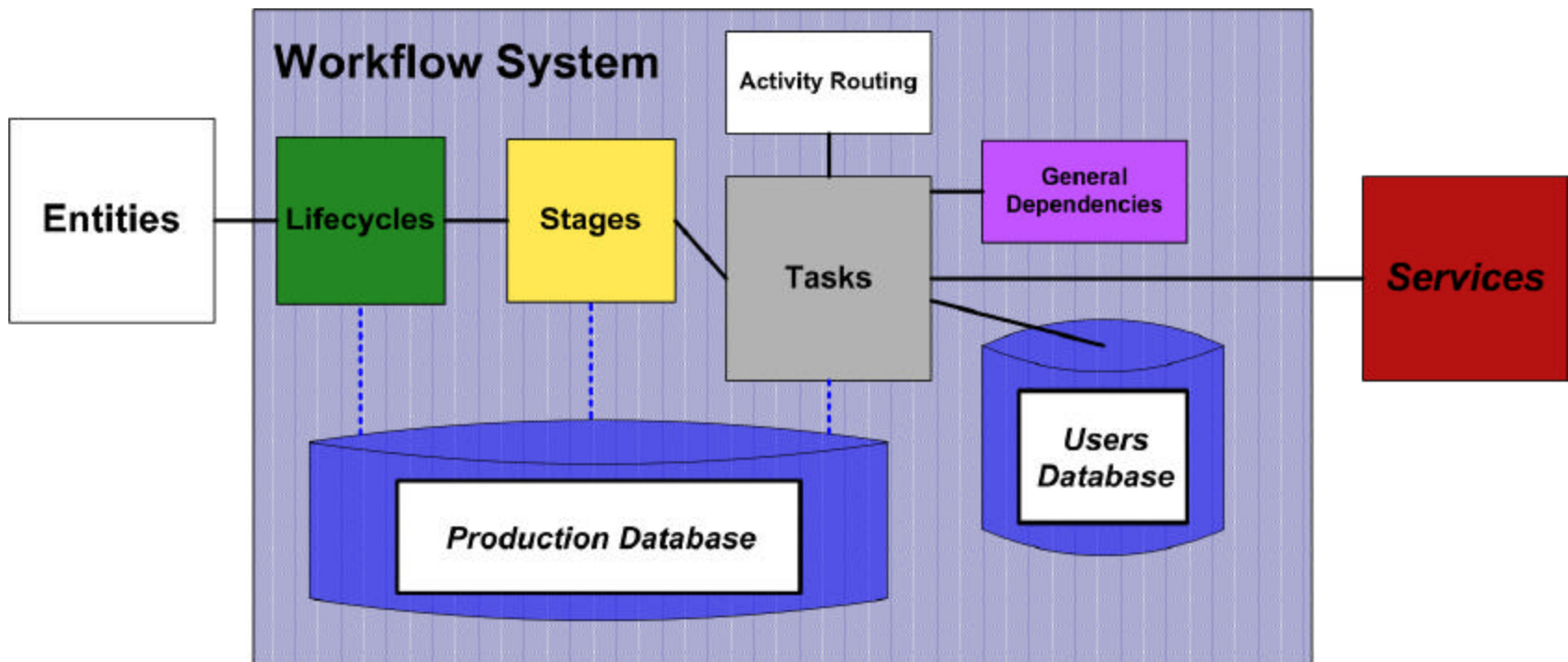
An Ex



Database Structure / Entity-Relation Model



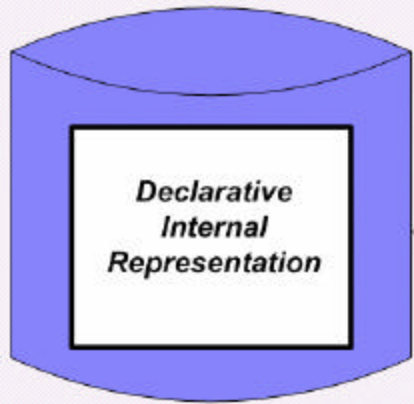
Overview of the Complete Model



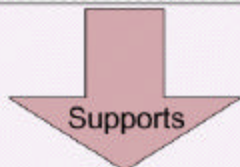
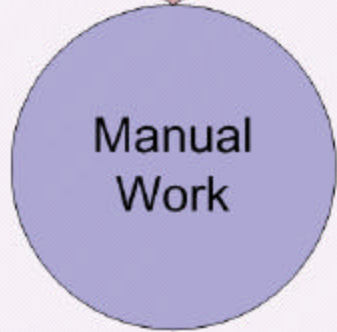
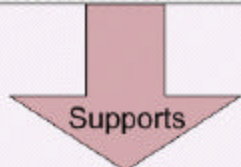
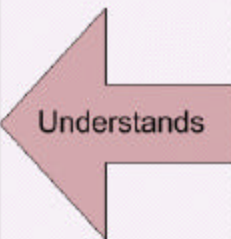
What is Missing from the E-R Model?

- The given structure is passive
- Nothing drives the activity
- A Workflow Engine is required to:
 - Ensure lifecycles proceed from stage to stage
 - Enact tasks
 - Enforce dependencies between tasks/lifecycles
 - Spawn child tasks and regulate task selection

Workflow System for Health



Workflow Data Repository



The Workflow Engine

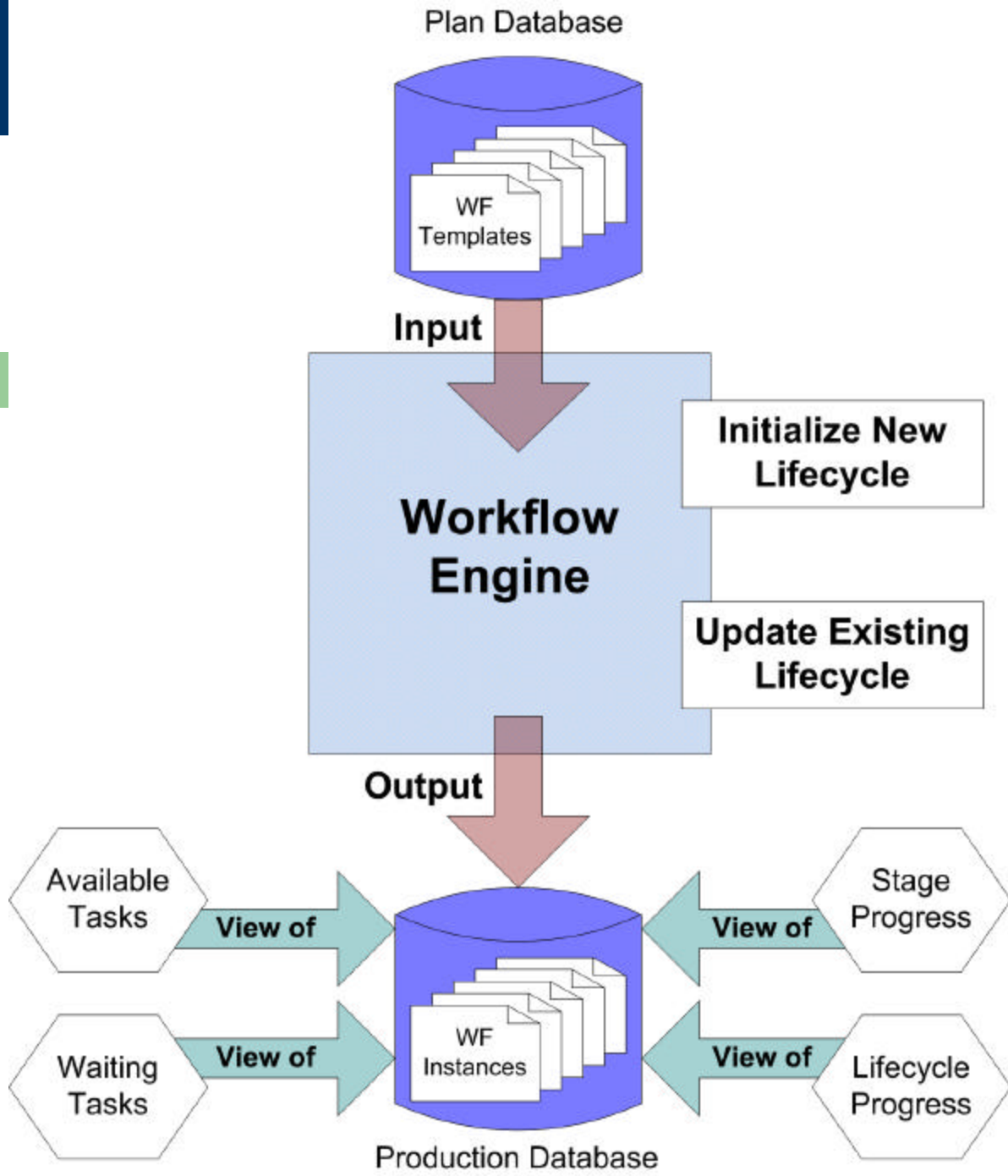
“WFE almost acts as a ‘higher-order’ operating system for application functionality”

~Lawrence-05

“The Engine is a piece of software that walks over the workflow model providing functionality at each step”

~Cowan-05

Implementing the Engine



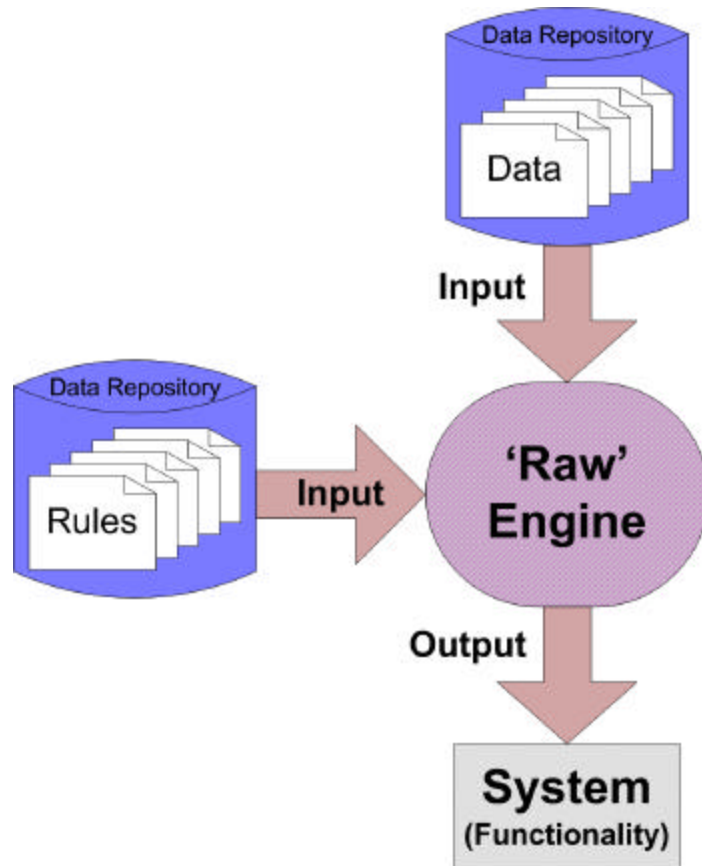
Implementing the Engine Using a Declarative Approach

“Data is more easily changed than programs”

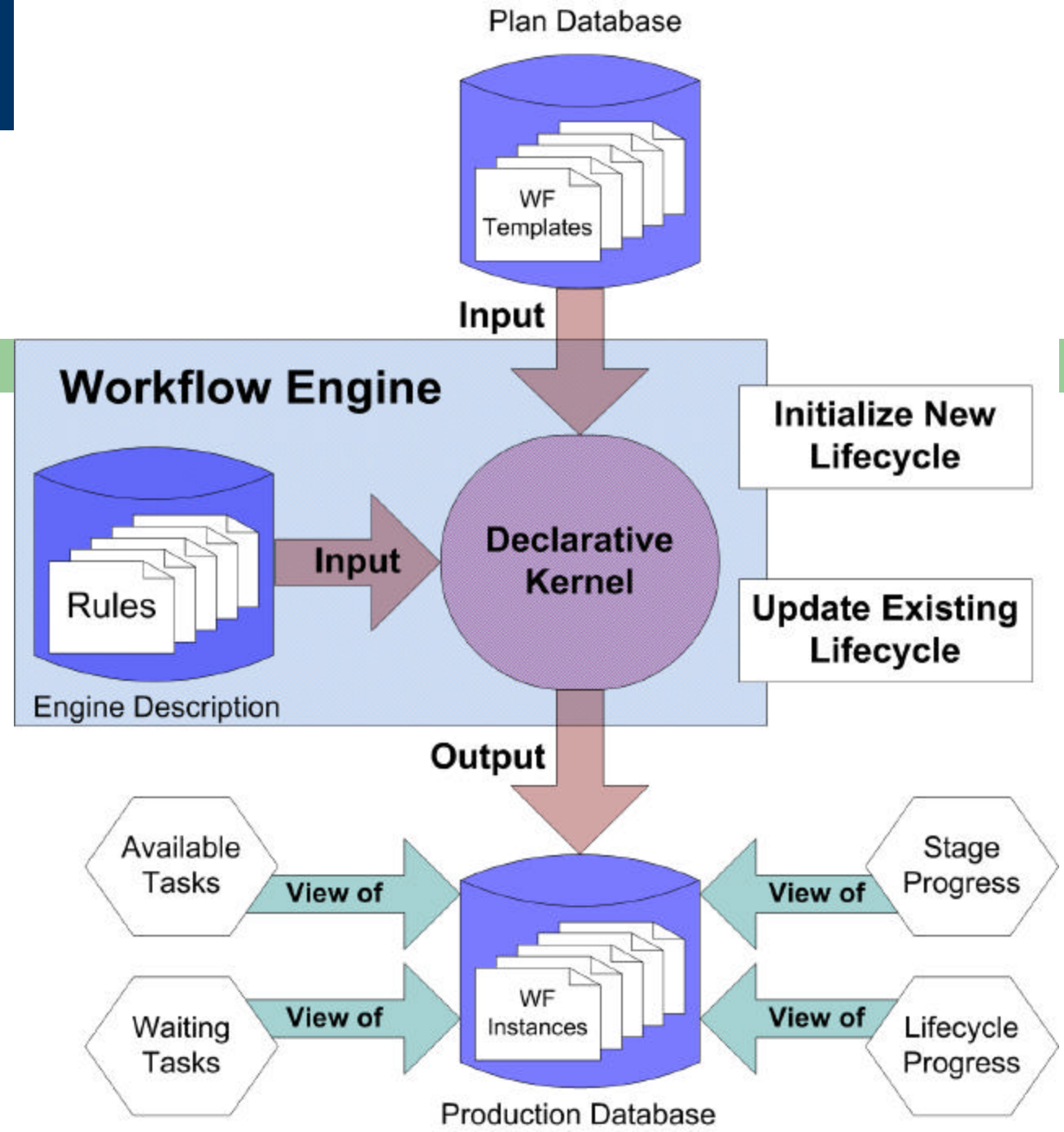
~Jackson-97

- Workflow is declarative by nature.
- We take this idea one step further and program the WFE using declarative methods.

A Raw Declarative System



- A purely declarative system would be built around an engine with little or no knowledge on its own (a raw engine)
- Rules of different types are fed into the engine before the 'program' determining the actions of the engine.
- In our case the workflow system is supplied with rules regarding routing, roles, and the treatment of contextual concerns.



“Declarative Kernel” - An Engine for the Engine

- The engine provides the following SQL procedures:
 - **Invoke(*behavior*, [*subject*])** – Fetches the *behavior* declared in the database and executes it immediately.
 - **Invoke_If(*condition*, *behavior*, [*subject*])** – Fetches the *behavior* from the database and executes it conditionally.
 - **For_All_Contained_Entities(*parentID*, *parent*, *child*, *behavior*)** – Fetches the *behavior* declared in the database and executes it for each child contained within the parent.

Invoking Behavior

- Behaviors are declared in terms of:
 - Other behaviors
 - Very basic SQL procedures.
- Generally behaviors declare what should occur, not how it occurs.
- Programming system behavior is reduced to an exercise in declarative composition.

Implementing the Workflow Engine

- The machinery of the declarative-system, along with a handful of basic SQL procedures and database triggers, provide all the pieces needed to implement the WFE.
- Specifically, the WFE functions **Initiate New Lifecycle** and **Update Existing Lifecycle** are implemented by declaring the behaviors they exhibit.
- Here the power of the declarative-system can be felt.

Limitations of the Implementation

- Even with the workflow engine the model is incomplete as a full workflow system:
 - The model simply regulates and constrains the firing of tasks.
 - It lacks 'scheduler' logic.
 - It is not yet able operable in a Web Services context.
- The workflow engine and supporting representations provide a sandbox for experimenting with workflow.
- To add functionality integration is required...

Integration with the WIDE Toolkit

- An environment for web-based systems
 - Supports declarative definition of service types:
 - Interactive reports
 - Interactive visual interfaces
 - Multimedia
 - Agents
 - Databases
 - ...
 - Semi-automatic build of workflow environment
-
- The Workflow Engine is embedded in WIDE as another service type.

Going Further... Providing Interface Capabilities

- One of the benefits of the declarative approach taken by this system is the ability to easily export tagged data.
- Two scenarios are immediately possible:
 - Prolog Analysis:
 - Views of workflow templates can be exported as Prolog fact sets
 - Combined with appropriate rule sets, workflows can be logically queried or analysed.
 - XSLT Transformation:
 - Workflow production data can be exported as XML files.
 - Workflow template data can be exported as XSD schema files.
 - Combined with appropriate XSLTs all manner of transformations are possible.

The Payoff...

- The ability to interface in this manner is a feature of the workflow engine.
- This allows:
 - Integration with the scheduling tools.
 - Integration with the dependency query-engine created by Weizhen Dai.
 - Workflow Mining:
 - Of increased value for Assembled workflow (mentioned earlier)
 - Workflow templates can be recovered / discovered through practise
 - Enables delta-analyses
 - Embedded Simulation

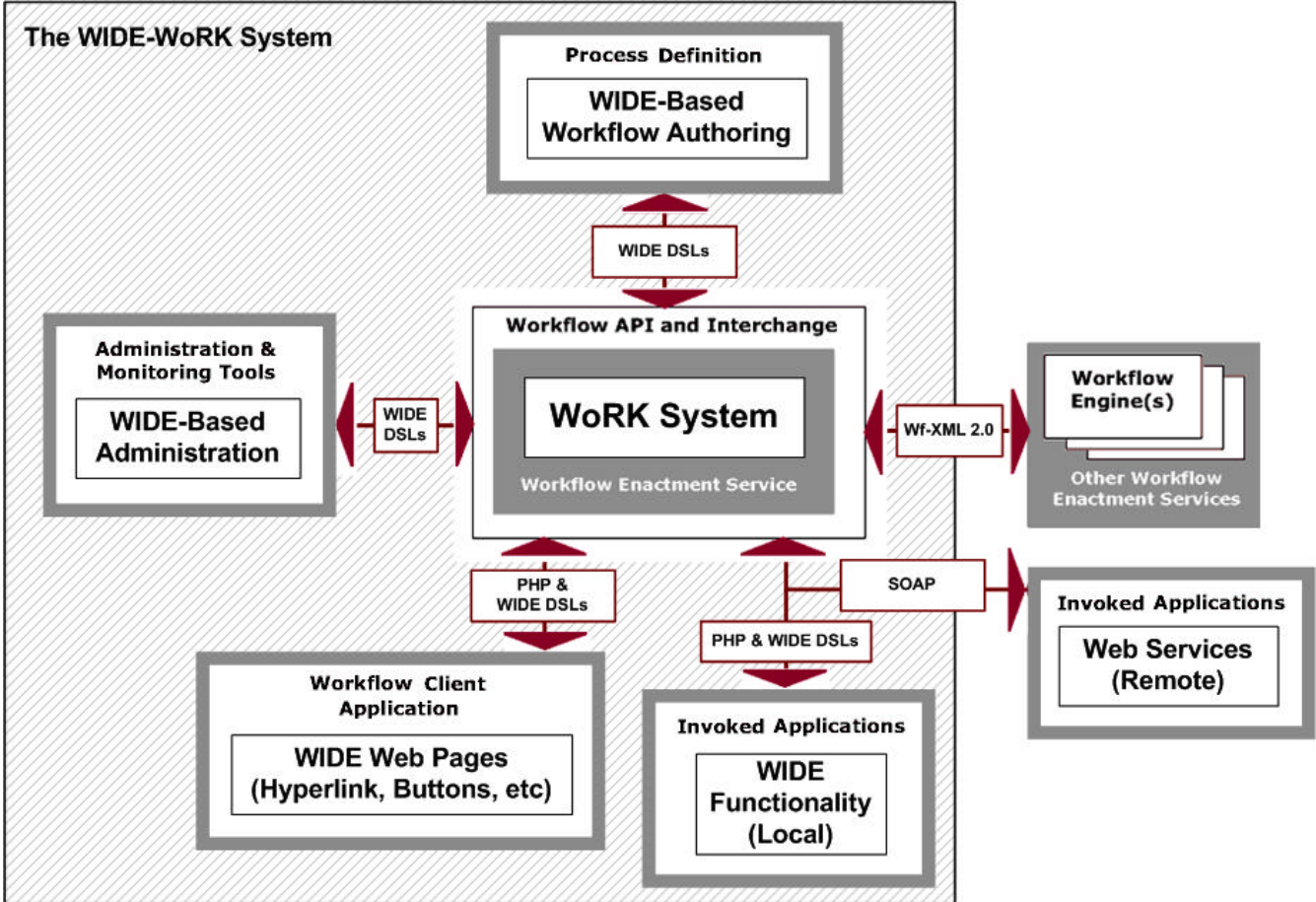
Evaluation

Evaluation

The integrated system was evaluated against:

1. The Workflow Management Collision Workflow Reference Model.
 - Describes the common features of Workflow / BPM systems.
 - Adherence to the model aids in achieving interoperability through WSDL.
2. The requirements for a modern workflow system for healthcare developed in the thesis.

The WIDE-WoRK System in terms of the Workflow Reference Model



The WIDE-WoRK System in terms of the Workflow Reference Model

- Three of the interfaces are already implemented.
- Providing the other two (through interoperability standards such as a SOAP interface) will produce full coherence with the model.
- This will allow the application of the cookbook approach to implementing a Wf-XML interface.

The WIDE-WoRK System in terms of the Requirements for a Modern Workflow System for Healthcare

- Results of this evaluation show that the workflow model at the core of the WoRK System meets many of the requirements directly.
- Evaluating the Model:
 - Without modification or expansion the model supports:
 - assembled and prescribed workflows
 - workflow routing
 - variable workflow granularity
 - rule-based workflow
 - a rich representation of agent roles.
 - The model is capable of capturing workflow patterns and representing an expanded notion of time.
 - The model can be expanded to include a rich representation of other contextual dimensions as needed by the workflow application domain.

The WIDE-WoRK System in terms of the Requirements for a Modern Workflow System for Healthcare

- Results of this evaluation show that the workflow engine meets, or has the capacity to meet, all the requirements specified.
- Evaluating the Engine:
 - Without modification or expansion the engine:
 - Understand the workflow representation
 - Can be easily extended declaratively
 - Functions in a local context supporting both automatic and manual workflows
 - The ability to export to XML-based and Prolog-based representations allows the engine to support:
 - Simulation
 - Workflow mining
 - Formal analysis

By interfacing with separate tools.
 - Finally, the engine corresponds with the Workflow Reference Model, which makes interoperability a possibility for future implementation.

Results

Results (so far...)

- The Workflow System:
 - Not complete, lacks: Scheduling, authoring, analysis and the model is missing some contextual concerns.
 - Some of these limitations are addressed by integration with WIDE
 - Others are met by XML and Prolog integration
 - The result: robust framework on which to build workflow solutions
 - Of particular benefit for healthcare workflow systems:
 - Explicit notion of Assembled vs. Prescribed WF
 - Rich treatment of User Roles
 - Handles Crosscutting Concerns*

Contributions

Contributions of this work include:

- A framework for evaluating workflow systems for the healthcare domain.
- The Blueprint for an Extensible Workflow System:
 - The two-tiered DSL for workflow facilitates an extensible workflow system.
 - Recipes are provided for extending the workflow model, and implementing any related behavior via the declaration of related behaviors.
 - Workflows expressed using the extended model can be enacted directly by the system without modification of the workflow engine.

Contributions

Contributions of this work include:

- A Variable Workflow Engine (Flexibility):
 - Behavior-sets can be swapped, leading to dynamic workflow behavior.
 - This design makes possible:
 - Running a workflow in different 'modes' (assembled vs. prescribed, standard vs. emergency)
 - Using the engine for workflow transformation (like XSLT)
 - Using the engine to simulate a workflow
- An experimental sandbox
 - Integration with WIDE provides system ideal for prototyping workflow tools.

Future Work

Future Work

- General:
 - Formal Treatment:
 - DSLs
 - Analysis
 - 'Prove some facts about Workflow'
 - Workflow Pattern Repositories
 - Workflow Engine Rule Repositories

Future Work

1. Aspect-Oriented Workflow
2. Middleware / Web Service Composition
3. Real-time Prediction of Assembled Workflow

Future Work - Aspects

- ***Aspect-oriented Workflow:***

- Extension of the workflow system to include concepts from aspect-oriented programming (AOP)

“Some workflow concerns are not easily captured by decomposition of a process into subprocesses and tasks. In fairly large and complex workflows concerns are sometimes tangled between tasks. One such workflow concern is logging. All tasks are logged in some fashion, however this logging can vary drastically between tasks even in the same workflow.”

- ***AWE - Aspect Workflow Engine***

Future Work - Middleware / Web Service Composition

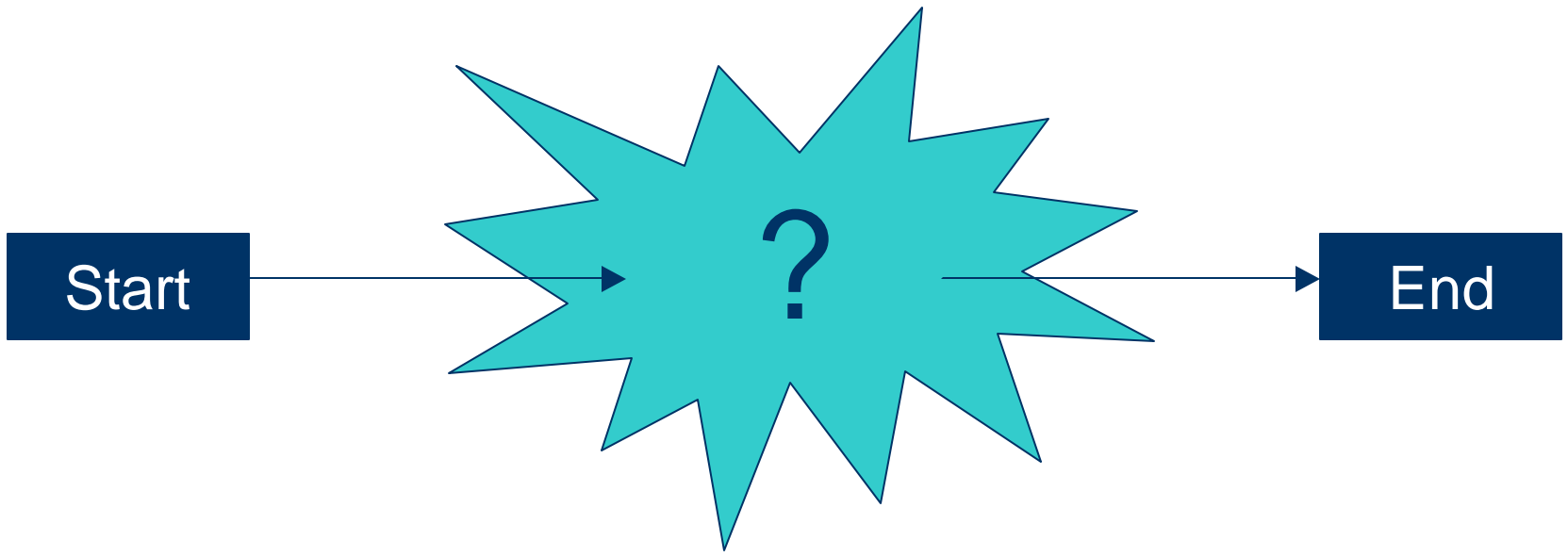
- Web Services Metamodels are used to tie together disparate services to construct virtual applications.
- Current metamodels typically mix different standards to implement a single metaprocess – (BPEL + WS-CDL + XPDL is a common mixture)
 - This leverages the best language for the job.
 - Leads to a complicated system at best (hard to analyze)
- Using the WIDE-System workflow model as a Web Services metamodel has advantages:
 1. The entire metamodel can be specified in a single system definition using only one common language (the DSL for Workflow).
 2. With a formal treatment of the DSL, facts can be proven about virtual enterprises that aren't possible when the definition of the enterprise stretches across various standards.
- Future research will focus on attempts to utilize the WIDE-WoRK System as a framework for implementing Web Service Metamodels. This is in pursuit of creating a better middleware solution.

Future Work - Real-time Prediction of Assembled Medical Workflow

- A real-time surgical guidance system.
 - The WIDE-WoRK System supports:
 - The representation of assembled workflow
 - The ability to perform workflow mining
 - Covvey suggests a system wherein surgery is captured as it occurs and the resulting assembled workflow is compared with previously mined workflow sequences (average courses of action taken) to form a series of hypothesis about what is occurring.
 - The most likely hypothesis is offered as advice to the surgeon
 - When/if the surgeon deviates from the workflow pattern, the next hypothesis is then presented.
 - In this application guidance regarding the process, or parts of it, can still be presented to the surgeon even though no prescribed workflow is provided. Such a real-time surgical guidance system is an exciting possibility but one that will require a great deal of further research.

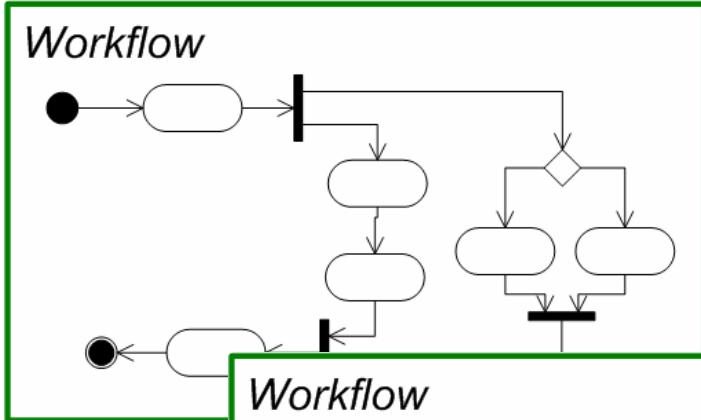
Open Problems

What is Dynamic Workflow?

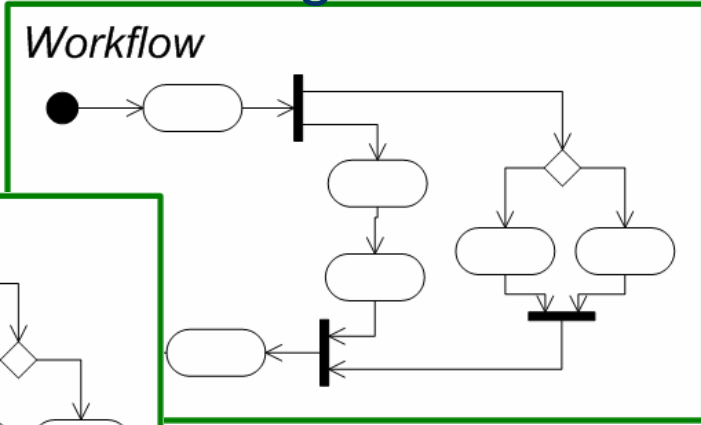


Multiple Flows...

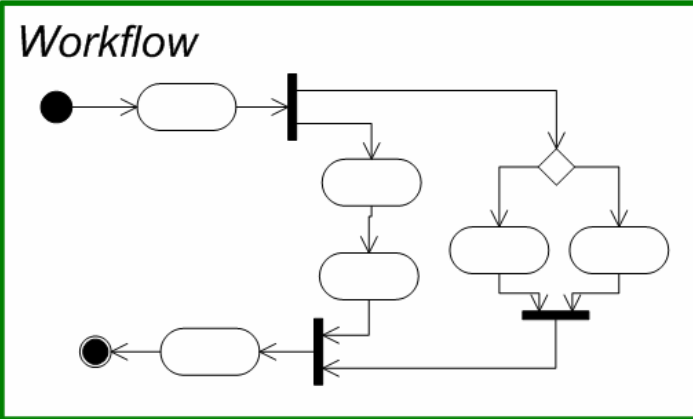
Actual



Cognitive



Workflow



Ideal

Crosscutting Concerns

- What are they?
- Are they important?
- Are aspects answer?

Wrap-up

Acknowledgements

- My supervisor - Don Cowan
- The WoRK Group (Agfa & NSERC):
 - Dominic Covvey
- Computer Systems Group

Questions



Thank you
for your attention.

bmalyk@csg.uwaterloo.ca

References (1)

- [AALST-98] W.M.P. van der Aalst. “The Application of Petri Nets to Workflow Management. of Circuits, Systems, and Computers”, 8(1): 21-66, 1998.
- [AALST-03] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns(PDF, 718 Kb) *Distributed and Parallel Databases*, 14(3), pp. 5-51, July 2003.
- [AGRAWAL-98] R. Agrawal, D. Gunopulos, F. Leymann. “Mining Process Models from Workflow Logs”, in Sixth International Conference on Extending Database Technology, 1998, pp. 469–483.
- [BAM] D. McCoy, Gardner Group. “Business Activity Monitoring – An Overview from Gardner Group”, a web cast viewable at: <http://whitepapers.zdnet.co.uk/0.39025945.60046972p-39000468q.00.htm>
- [BPEL] IBM’s website for Business Process Execution Language – <http://www-128.ibm.com/developerworks/library/ws-bpel>.
- [BPMI] Website for the Business Process Management Initiative – <http://www.bpmi.org>.
- [BRG] Website for the Business Rules Group – <http://www.businessrulesgroup.org>.
- [CHI] Website for the Canada Health Infoway – <http://www.infoway-inforoute.ca>.
- [COVVEY-PC-06] H.D. Covvey, *private communication*. 2006.
- [COWAN-PC-05] D. Cowan, *private communication*. 2005.
- [COWAN-06] D.D. Cowan, P.S.C. Alencar, D.B. Brown, H.D. Covvey, I. Gimenes, C.J.P. Lucena, W.J. Malyk. “Composition of Web-based Services Using a Service-oriented Model based on Workflow Concepts” (*In Preparation March 2006*).
- [DAI-PC-05] W. Dai, *private communication*. 2005.
- [DAI-05] W. Dai, “A *Query-Based Approach to Workflow Process Dependency Analysis*”. MMath Thesis, University of Waterloo, 2005.

References (2)

- [DUBRAY-05] J.J. Dubray. “Business Process Metamodels and Services” from *The Workflow Handbook 2005*, Future Strategies Inc., 2005, ISBN 0-9703509-8-8, pp. 159-178.
- [DUBRAY-99] J.J. Dubray et al. “[An extensible object model for business-to-business e-commerce systems](#)”, OOPSLA 99, Business Object Component Workshop.
- [ELRAD-01] T. Elrad, R. Filman & A. Bader. “Aspect-Oriented Programming: Introduction”, *Communications of the ACM*, 2001, vol. 44, #10, pp. 28-32.
- [FIELD-92] M.J. Field and K.N. Lohr Eds. Institute of Medicine. “Guidelines for Clinical Practice” from *Development to Use*, National Academy Press, Washington D.C., 1992.
- [GAMMA-95] E. Gamma, R. Helm, R. Johnson, J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 95, ISBN: 0-201-633612.
- [GRECO-04] G. Greco, A. Guzzo, G. Manco, L. Pontieri, D. Saccà. “Mining Constrained Graphs: The Case of Workflow Systems”, Italy, 2004.
- [GROK] Website for the Grok Interpreter – <http://swag.uwaterloo.ca/~nsynytksyy/grokdoc/grokintr.html>.
- [HOLLINGSWORTH-04] D. Hollingsworth. “[The Workflow Reference Model: 10 Years On](#)” from *The Workflow Handbook 2005*, Future Strategies Inc., 2004, ISBN 0-9703509-6-1, pp. 295-312.
- [HOLT-02] R. Holt, *TA: Tuple Attribute Language*. 2002. <http://plg.uwaterloo.ca/~holt/papers/ta-intro.htm>.
- [HL7] Website for Health Level 7 – <http://www.hl7.org>.
- [JACKSON-97] M. Jackson, G. Twaddle, *Business Process Implementation – Building Workflow Systems*. Addison-Wesley, 1997, ISBN: 0-201-177684.
- [KARRELS-99] http://www.karrels.org/Ed/ACM/98/prob_e.html.
- [KICZALES -97] G. Kiczales et al. “Aspect-Oriented Programming”, ECOOP, 1997.
- [LAWRENCE-05] C. Lawrence. “Integrated Function and Workflow” from *The Workflow Handbook 2005*, Future Strategies Inc., 2005, ISBN 0-9703509-8-8, pp. 31-52.

References (3)

- [MALYK-06] W. Malyk, “Towards an Extensible Workflow System for Healthcare,” master thesis, University of Waterloo, 2006.
- [MCKAY-PC-06] K. McKay, *private communication*. 2005-6.
- [MCKAY-05] K. McKay, presentation: “Understanding Temporal Context Dependency In Production and Work Flow Scheduling”. September 14, 2005.
- [MARUSTER-01] L. Maruster, W.M.P. van der Aalst, A.J.M.M. Weijters, A. van den Bosch, W. Daelemans, “Automated discovery of workflow models from hospital data”, in: B. Kröse, M. de Rijke, G. Schreiber, M. van Someren (Eds.), Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001), 2001, pp. 183–190.
- [MIERS-05] D. Miers. “BPM – Too Much BP, Not Enough of the M” from *The Workflow Handbook 2005*, Future Strategies Inc., 2005, ISBN 0-9703509-8-8, pp. 23-30.
- [MILNER-99] R. Milner. *Communicating and mobile systems: π -calculus*. Cambridge University Press, 1999, ISBN 0-521-65869-1.
- [PLESUMS-05] C. Plesums. “Workflow in the world of BPM: Are They The Same?” from *The Workflow Handbook 2005*, Future Strategies Inc., 2005, ISBN 0-9703509-8-8, pp. 17-22.
- [SOAP] D. Box et al. “Simple Object Access Protocol (SOAP) 1.1”, W3C Note, May 8, 2001.
- [SOC] M. Papazoglou et al. “Service Oriented Computing”, Communications of the ACM, October 2003, Vol. 46 No. 10, p25.
- [STEINER-05] J. Steiner by way of D. Covvey, 2005.
- [SWENSON-05] K.D. Swenson. “ASAP/Wf-XML 2.0 Cookbook - Updated” from *The Workflow Handbook 2005*, Future Strategies Inc., 2005, ISBN 0-9703509-8-8, pp. 257-280.
- [SW] Website for the Semantic Web – <http://www.w3.org/2001/sw>.
- [QUAGLINI-01] S. Quaglini, M. Stefanelli, G. Lanzola, V. Caporusso, S. Panzarasa “Flexible guideline-based patient careflow systems”, Artificial Intelligence in Medicine, Volume 22, Issue 1, April 2001, pp. 65-80.
- [UML] Website for the Unified Modeling Language – <http://www.uml.org>.

References (4)

- [WFMining-03] W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, J. Maruster, G. Schimm, A.J.M.M. Weijters, “Workflow Mining: A Survey of Issues and Approaches”, Internal Report, 2002.
- [WfMC] Website for the Workflow Management Collision – <http://www.wfmc.org>
- [WfMC-REF] Website for the Workflow Management Collision Workflow Reference Model – <http://www.wfmc.org/standards/model.htm>
- [Wf-XML] Website for Workflow-XML – http://www.wfmc.org/standards/wfxml_demo.htm
- [WHITE-04] S. White et al. “business Process Modeling Notation v1.0”, May 2004, BPML.org.
- [WS] Website for the "Web Services Activity Statement" - <http://www.w3.org/2002/ws/Activity>
- [WS-CDL] N. Kavantzias et al. “Web Services Choreography Description Language Version 1.0”, W3C, Working Draft 27 April 2004.
- [WSDL] R. Chinnici et al, “Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language”, W3C, Working Draft 3 August 2004.
- [XML] Website for the Extensible Markup Language – <http://www.w3.org/XML>
- [XPDL] Website for the XML Process Definition Language – <http://www.wfmc.org/standards/XPDL.htm>
- [YAWL] Website for Yet Another Workflow Language – <http://www.yawl.fit.qut.edu.au>