

Micro-Workflow: An Object-Oriented Workflow Architecture for Embedded Workflow

Dragoş A. Manolescu
Department of EECS
University of Kansas
manolesc@ittc.ku.edu

August 24, 2001

Background: Research Focus

- Software patterns
- Object-oriented frameworks
- Workflow management

Software Patterns

- Data flow and multimedia
- Information retrieval
- eBusiness

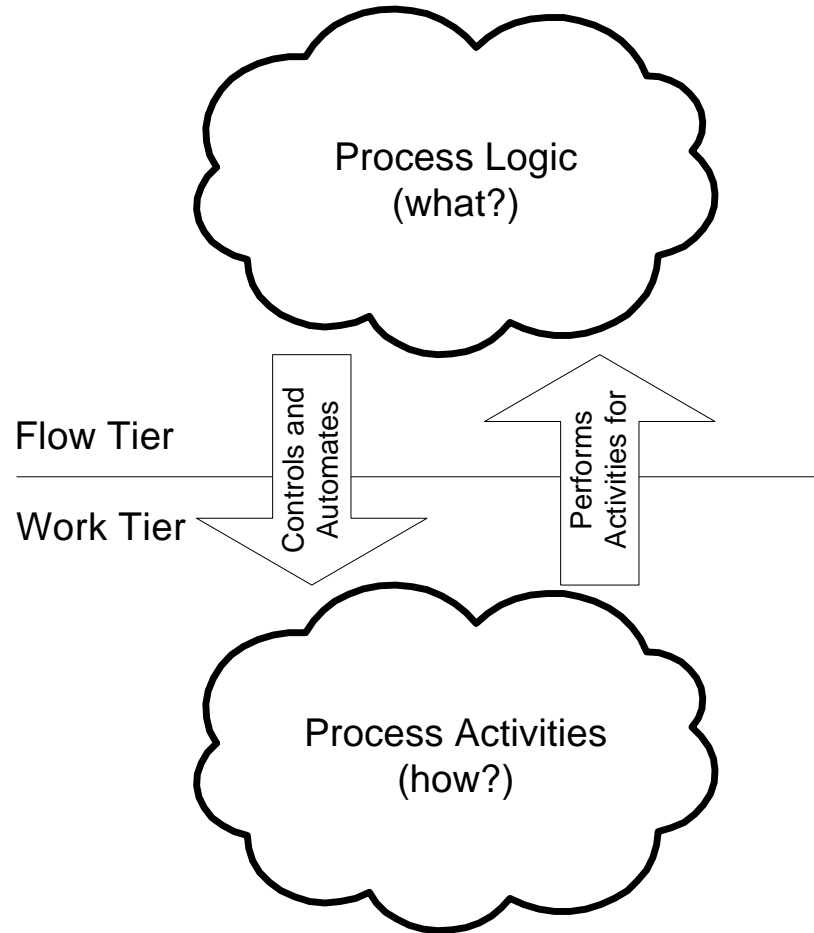
Object-Oriented Frameworks

- Telecommunications billing
- Patient tracking and object-relational persistence
- eBusiness (stateful Web-based applications, customer care, presentation)

Workflow Management

- Object-oriented process and product models
- Patterns for workflow management
- Workflow architectures
- Object-oriented workflow frameworks
- Micro-workflow

What is Workflow?



Workflow Application Domains

Workflow for process-centric applications:

- (e)Business
- Insurance
- Financial
- Manufacturing
- Scientific

Micro-workflow: Motivation

- Three object-oriented frameworks: Objectiva (telecommunications billing), the Hartford Insurance Framework, and the Argo framework (school administration)
- All three implement processes with workflow
- The architects used a **custom** workflow solution
- Why not use one of the 200-300 **existing** workflow systems?

Workflow in Object-Oriented Applications

Typically application objects provide functionality along four dimensions:

What to do	Process logic
When to do it	
Who does it	
How to do it	Task logic

Implement the process logic with a workflow system; the application becomes **flow-independent** [Leymann+2000]:

- Separates concerns
- Change one without changing the other
- Simulate (before) and analyze (after) the process
- Use other workflow features

The Research Problem

Evolution: late 70s, office automation (e.g., OfficeTalk); early 90s, business processes (e.g., the Coordinator); late 90s, middleware (OMA's Workflow Management Facility).

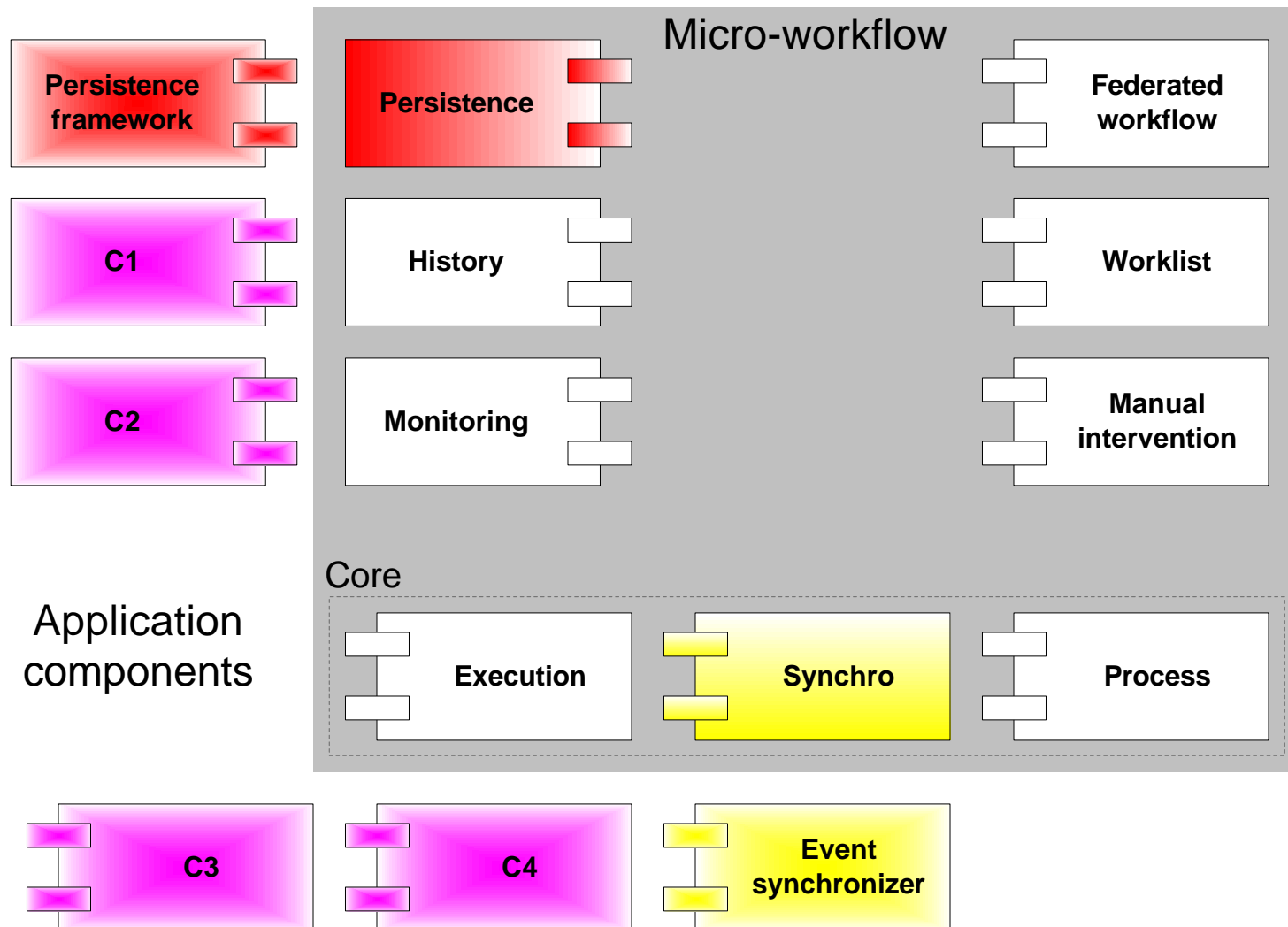
Current workflow systems

- Target non-programmers
- Focus on packaging many features
- Regard objects as an implementation technique
- All-or-nothing solution
- Monolithic, heavyweight architectures

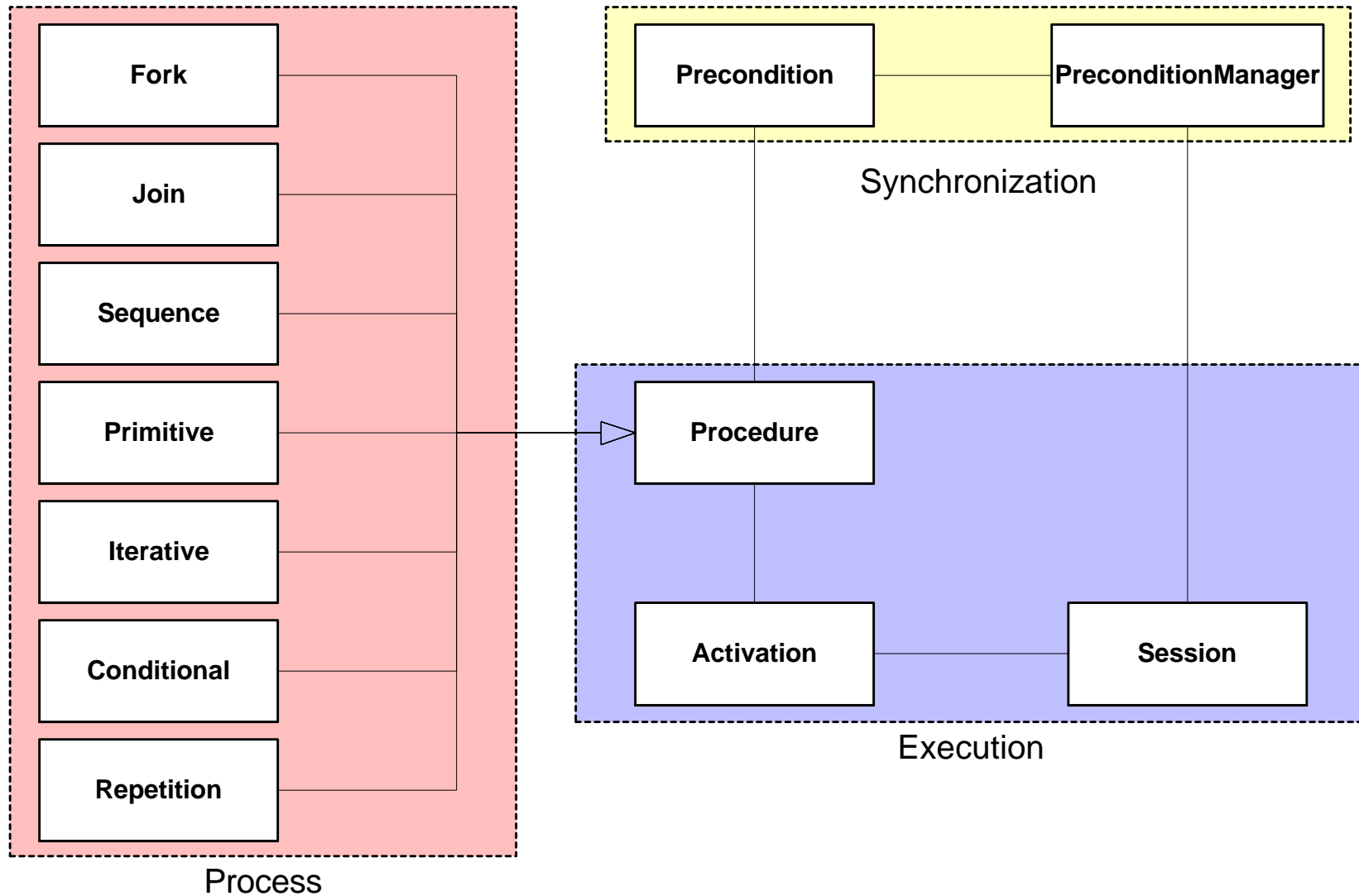
Embedded workflow

- Target programmers
- Full control over the features
- Integrate with applications and systems
- Gradual transition
- Customizable, reusable architectures

The Solution: Micro-Workflow



Workflow Core



Micro-workflow Components

Add workflow features by plugging in components

Component	How to add to the core
History	Plug a logging strategy into the workflow session
Persistence	Select a persistent logging strategy; plug a database session manager into the workflow session
Monitoring	Register a procedure monitor as a dependent of the workflow session
Manual intervention	Plug a rewinder class into the workflow session
Worklist	Replace domain objects with worklists
Federated workflow	Use SubworkflowProcedure on a facade; register the facade with the name service

Implementation

Micro-workflow framework (Smalltalk, Opentalk, GemStone/S)

Have built three applications:

- Administrative process (NCSA allocations)
- Patient management and tracking
- Newborn followup (Illinois Department of Public Health)

Summary

- Workflow is moving from an end-user application to middleware services
- Current workflow architectures are based on requirements and assumptions that make them unsuitable for embedded workflow
- Micro-workflow solves the problems of current workflow architectures in the context of object-oriented software development

Coming Soon

- *Several Patterns for eBusiness Applications*, accepted at PLoP 2001 (Monticello, IL)
- *An Extensible Workflow Architecture with Objects and Patterns*, in review for TOOLSEE 2001 (Varna, Bulgaria)
- *Why Java is Not Suitable for Object-Oriented Frameworks* (poster), accepted at OOPSLA 2001 (Tampa Bay, FL)