Weak Workflows in FRODO TaskMan
– System Walkthrough and Evaluation

German Research Center for Artificial Intelligence
Knowledge Management Research Department
Kaiserslautern, Germany
Remarks

- one result of the FRODO (Framework for distributed organizational Memories) project (Duration 2001-2003)
  - see http://www.dfki.de/frodo

- FRODO TaskMan is a prototype which is still under development

- The system is used to show important ideas – more to come ...

- The system will be used also in the successor project EPOS (Evolving Personal to Organizational Knowledge Spaces)
  - see http://www.dfki.de/epos

- state June 2003
Agenda

- System Architecture
- TaskMan Walkthrough
- Evaluation
FRODO WWf-System Architecture Supports Knowledge-Intensive Work by an Agent Society
Repository agents manage the workflow-relevant knowledge about models and structures

- The *organisational repository* manages enterprise-related information
  - users, departments, groups, projects, roles, competencies, skills, & their relationships
  - user profiles containing user information, skills, experience

- The *task concept ontology* offers the concepts to describe tasks
  - ontology browsing supports modelling in finding appropriate task models, resp. ‘building blocks’ for a user’s current tasks

The *model repository* acts as an information source for
  - providing ‘blue prints’ of tasks
  - providing ‘building blocks’ for workflow modelling

- The *audit repository* logs workflow actions within the workflow system
  - includes the audit trail of task instances, state changes, modifications...
  - observes the history of task instances and changing models
  - answers questions about former instances of a task (e.g. to find best/worst practice)
The WWf Core Agents enact the processes

- **YPA** registers information about agent responsibilities
  - "TIA \( n \) responsible for task instance \( X \)"

- **WIPA** assigns task instances to users
  - according to the process role
  - keeps track of active users, new/removed work items, informs user agents about work items

- **Task Instance Agents** TIA have the goal to get their task instance done,
  - e.g., register it in the WIP, listen for event changes of other TIA in order to get started, or acquire necessary input data
  - the overall workflow is done by cooperating TIAs each responsible for a subtask
  - TIAs are created by TI Generator Agents

- **User Agents**
  - represent a specific user within the workflow system
  - support the knowledge work by interacting with different information sources
Ontology Agents and Information Agents cooperate to satisfy information needs

- **Domain / Information Ontology Agents**
  - manage their respective ontology
  - provide typical ontology services
  - assume multiple roles in a distributed, cooperative scenario
  - offer the basis for information structuring

- **Information agents** realize information access and delivery
  - make information objects accessible to the agent system (esp. wrapper agents for legacy systems)
  - rely on ontologies to structure and access information objects
  - satisfy information needs specified by User- / task Instance Agents

- **All agents cooperate to perform distributed inferences for information retrieval**
Example: A task instance reaches its responsible user, seen in a step-by-step communication

- TIA registers the responsibility for its task instance
- TIA registers process role at the work item pool
- if user is active, the user agent is informed about the new work item (task instance id)
- to get the task instance data, YPA is asked for the TIA responsible for the given id
- YPA answers with the agentID of the TIA
- UserA requests task instance data from TIA
- task instance is now available in the user manager
Agenda

- System Architecture

- TaskMan Walkthrough

- Evaluation
FRODO TaskMan Walkthrough

- UserManager Interface
- Modifying the Workflow Model
- Modelling & Executing a Task
- Working with Information Objects
FRODO TaskMan is used during the user’s everyday work

- a tell tale window informs the user about news from tasks he is involved
- now, the user can access his UserManager
The Worklist shows task the users is involved in

user’s workitems

an organisational view on the worklist

all workflows with open tasks the user is involved
The graph view shows the workflow model

- a finished task
- successor relation
- task has 3 (hidden) children
- task in progress and currently selected (red line)
- task stays initiated until predecessor is finished
- the workflow (resp. root task)
The graph allows task navigation showing also subtasks and interdependencies.

- A processible task
- One subtask
- Dotted lines show subtask relation
An integrated task manager enables the user to get informed about a task and to model and work.
During its lifetime, a task instance has different states:

- **start of each task instance**
  - wait until preconditions are fulfilled

- **the task is ready to be processed**
  - a user is executing the task

- **task is processible, but execution temporarily paused**
  - ‘put back on stack’

- **task is completed**
  - task is completed

- **task is suspended**
  - task is suspended

- **terminate**
  - task will be deleted

- **terminate**
  - terminate the task

- **initiated**
  - initiated task

- **processable**
  - processable task

- **inProgress**
  - task in progress

- **completed**
  - task completed

- **suspended**
  - task suspended

- **terminate**
  - terminate the task

- **outQueued**
  - task sent to queue

- **queueOut**
  - task queued

- **reinit**
  - task reinitiated
The task control bar

- requests modelling
- requests work
- commit changes
- revert changes
- finishes a task
- modelling assistant (guided interface)
- task navigation history

path to root (workflow) allows direct access to supertasks

current state

To access supertasks, use the task control bar. It includes options for task management: requesting modelling, requesting work, committing changes, reverting changes, and finishing a task. The modelling assistant provides a guided interface for tasks. The path to root enables direct access to supertasks. The current state is displayed in the task path.
FRODO TaskMan Walkthrough

- UserManager Interface
- Modifying the Workflow Model
- Modelling & Executing a Task
- Working with Information Objects
Refinement during runtime via the modelling assistant: adding subtasks which are in sequence
This results in two tasks added which are ready to be executed.
Defining successors

draw a line from a task to its successor

task changes state to initiated
Modification means also removing edges and deleting tasks

state again processible

allowed only for task leaves
FRODO TaskMan Walkthrough

- UserManager Interface
- Modifying the Workflow Model
- Modelling & Executing a Task
- Working with Information Objects
Modelling a task

- There is a logical difference between just modelling a task vs. working on it (e.g., finish only in work available).
- Changes to the task model are allowed: informal description, process role, information objects, ...
- Because of this, we depend on the user’s discipline not to work if he just requested modelling.
in the work-mode task model changes are also allowed

Put back, means that the users has temporarily stopped working on this task, the task is unlocked and can be requested by others

the state changes to **active**: the task is processible again, however, this reflects that someone has already done some work
Finishing a task enables successors to be processible

- remember, the system is flexible, therefore, work is still allowed
- other state changes can be done via an admin console (right mouse click on work button)
The worklist reflects these changes and allows further investigation by default, only ‘interesting’ states. Now, other tasks can be accessed.
Specifying who actually should do the work

- A process role defines the potential workflow participants.
- We allow:
  - User
  - Department, group, project
  - Skill
  - Role
  - Experience
- In the combinations and, or, not.

All members of the project FRODO could execute this task.

Now restricted to researchers in FRODO.
The worklist reflects this after committing the changes.

This user is a **FRODO member** and has the role **scientist**.

Now restricted to researchers in FRODO.
FRODO TaskMan Walkthrough

- UserManager Interface
- Modifying the Workflow Model
- Modelling & Executing a Task
- Working with Information Objects
Attaching information objects to a task

In the information tab several objects can be added:

- keywords, specified by the user
- documents (by URLs)
- concepts from domain ontologies

Provide/collect relevant information

Describe the task’s content; also used for information delivery

Available domain ontologies
The user can also create memos

- memos allow to attach relevant textual information to a task
- the author is logged
- different message types for the memo are available
- the memos will be a part of the OM (as process knowledge)
The TaskMan gives task specific information support in a browser tab.

This comprises:

- Attached documents, domain concepts, keywords, memos
- Inherited information from supertasks
- Attached information need
- Attached & inferenced task concepts

Currently implemented:

- Internet search with concepts
- Find related pages (via Google)
- Related documents in the organisational memory (via MindAccess™ from Insiders Inc.)

Inherited from super tasks:

- Related search
- Internet search
An integrated task-enabled web browser allows task specific handling of documents

Information Support for Task: Related work

Relevant Information
- Helko Mans' Workflow Management Links
- WFMC - The Workflow Reference Model

? Information Need
- combined search
- weakly structured workflows
- weak workflow
- Organisational Memory
- knowledge intensive work
- Workflow Management
- knowledge intensive tasks

search Organisational Memory

Mozilla webclient

browse to the results page of the information agent
An InfoAgent has found relevant documents in the OM domain concepts & keywords attached to task. Results listed with respect to their content.
Documents can be added to the task

if modelling is allowed, the browser offers to add the document to the task

Integration of a DAU-System in Workflow Management Systems

example is presented to understand the use. A conclusion completes the paper.

2 DAU

As mentioned above DAU aims at extracting information from paper sources and translating it into a structured representation. By this the

Information Support for Task: Related work

- (fulltext) Heiko Maas' Workflow Management Links
- (fulltext) Integration of a DAU System in WfMS
- (fulltext) WFMC - The Workflow Reference Model

Milestone 1
Agenda

- System Architecture
- TaskMan Walkthrough
- Evaluation
The experimental evaluation concentrates on core aspects of FRODO

- The experiments shall test whether
  - Weak Workflows are a useful basis for support of knowledge-intensive activities
  - Integration of process execution and information support is accepted and considered of benefit
  - Process-embedded information is a means of knowledge sharing and transfer
We chose visit planning as the knowledge-intensive activity to be supported

- Scenario 1: Prepare a visit for a prospective student to get familiar with the future work/living environment
- Scenario 2: Prepare a visit for a guest professor at the university, including tourism aspects

- The scenarios include room for flexible interpretation
- Ultimately a time schedule needs to be produced
- Both scenarios can be understood and processed in reasonably short time
- There is plenty of relevant information in the web
Both scenarios lead to structurally similar initial workflows

**Requirements analysis**

- **Student scenario**
  - Shorten
  - Concentrate on relevant activities
  - Search about location & environment

- **Professor scenario**
  - Search relevant activities
  - Commit & plan

---

Search relevant activities
(information about location & environment)
Four key elements of FRODO were employed in the evaluation trials

- Expressive power of the workflow language:
  - hierarchical decomposition
  - sequential dependency
- Dynamic refinement of workflows at execution time
  - integration of modeling & enactment
- Enriching workflow tasks with information needs
  - dynamic & static
- Support of task execution by linked information items
  - context-specific information support

The system demo illustrates these topics.
The Experimental Design Comprised Four Factors

- **Domain**: Preparing a schedule for a professor in Saarbrücken vs. for a student in Munich
- **Flexibility mode** of the workflow: Strict vs. weakly-structured
- **Complexity** of given workflows: “small” vs. “big” model
- **Re-use of information**: With/Without Given Information Items

A full (2x2x2x2)-factorial design was not feasible. This led to a restricted design with two experimental groups and the following features:

- Both groups process both **domains** and both **flexibility modes**.
- Complexity and re-use of information is tested only in one domain and with flexible workflows (between groups).
The experiment comprises the model-work-refine phases of the workflow lifecycle

Phase 1: Practice

- **Task:** Model
- **Domain:** Professor
- **Flexibility Mode:** -
- **Model Complexity:** -
- **Given Info Need:** -

Phase 2: Work I

- **Task:** Work
- **Domain:** Student
- **Flexibility Mode:** Strict
- **Model Complexity:** Low
- **Given Info Need:** -

Phase 3: Work II

- **Task:** Work
- **Domain:** Professor
- **Flexibility Mode:** Flexible
- **Model Complexity:** High
- **Given Info Need:** YES

Group 1

- **Task:** Model
- **Domain:** Professor
- **Flexibility Mode:** -
- **Model Complexity:** -
- **Given Info Need:** -

Group 2

- **Task:** Model
- **Domain:** Student
- **Flexibility Mode:** -
- **Model Complexity:** -
- **Given Info Need:** -

- **Task:** Work
- **Domain:** Professor
- **Flexibility Mode:** Flexible
- **Model Complexity:** Low
- **Given Info Need:** -

- **Task:** Work
- **Domain:** Student
- **Flexibility Mode:** Strict
- **Model Complexity:** Low
- **Given Info Need:** -
The evaluation trials gathered direct and indirect measurements

- direct measurements: The test persons were asked for subjective assessments via questionnaires
- indirect measurements: data collected during the experiment were evaluated
  - modified workflow models
  - attached information items
  - web logs, representing search activities

We performed 5 trial runs with 25 students in total. The first run was considered a pre-study.
Four hypotheses were proven by the experiment

- Knowledge workers feel better supported with late/lazy modeling facilities
- Lazy/late modeling in weak workflows leads to a more precise classification of information items than strict workflows
- Proactive information support is (demonstrably) useful
- Weak workflows are better suited than strict ones to deal with unexpected task situations
Evaluation of the questionnaires proves: Knowledge workers feel better supported with late/lazy modeling facilities

- All groups appreciate the flexibility of weak workflows
- Appreciation is significantly higher in the groups who started with the weak workflows
  - people appreciate the benefits after they lose them

The inverse question gave an identical result.
Analysis of the modified workflow models shows:
Dynamic modeling is used intensively.
Analysis of enriched workflow models demonstrates: Lazy/late modeling in weak workflows leads to a more precise classification of information items than strict workflows.

- weak workflows result in less information items / task
- this is interpreted as a more precise classification
Proactive information support is (demonstrably) useful

Evaluation of web access logs shows:

- About 30% of all information access result from pre-given information
- User profit from pre-given information elements
- Nevertheless, additional information sources are visited
Weak workflows are better suited than strict ones to deal with unexpected task situations

- After 1 hour’s work, an additional task was introduced in each case:
  - weak workflow: “The professor indicates that his wife intends to do sightseeing & wellness. Check possibilities and make relevant suggestions”
  - strict workflow: “The student intends to earn money by giving music instructions. Check possibilities and contacts”

- The analysis of the work results shows the advantage of weak workflows:
  - weak workflow: the task is integrated into the process
  - strict workflow: the task is
    - partially ignored
    - wrongly classified
Unexpected task in a weak workflow: Sound integration of the additional task and information related to it.
Unexpected task in a strict workflow: Relevant information is (inadequately) linked to ‘town’
The results of the experiment answer further interesting questions

• Does pro-active information support hinder creativity?
  – No: Data show that more information is processed and innovative solutions can be carried out

• Can process models form a part of shareable individual and organizational knowledge?
  – Yes: The knowledge transfer via improved process models could be demonstrated
The experimental evaluation was harder than expected but delivered favorable results

- Long preparation
  - sound design of an experiment was difficult
  - sound example proved hard to construct
  - design, performance and evaluation were time-consuming
  - first experiment runs discovered previously unknown deficiencies in various tool implementations

- Benefits of the FRODO approach were clearly demonstrated
  - Hypotheses were positively proven
  - Further questions could be answered

The analysis of the experimental results continues. A publication is submitted.
More questions ... ?

Contact:
Heiko.Maus@dfki.de
Sven.Schwarz@dfki.de

Evaluation:
elst@dfki.uni-kl.de

Website
http://www.dfki.de/frodo

recent publication

**Weakly-structured Workflows for Knowledge-intensive Tasks: An Experimental Evaluation**

Ludger van Elst, Felix-Robinson Aschoff, Ansgar Bernardi, Heiko Maus, Sven Schwarz
in Knowledge Management for Distributed Agile Processes: Models, Techniques, and Infrastructure (KMDAP2003) at WETICE-03

http://www.dfki.uni-kl.de/~maus/publ.html#WET-ICE03

FRODO TaskMan Version April 2003