Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 34 (2008) 551–563
Expert Systems with Applications
www.elsevier.com/locate/eswa

An expert system for dynamic re-coordination of distributed workflows ☆

William L. Kuechler Jr. [a,*], Vijay K. Vaishnavi [b]

[a] *Department of Accounting and Information Systems, University of Nevada at Reno, United States*
[b] *Department of Computer Information Systems, Georgia State University, United States*

Abstract

A persistent problem in the use of automated workflow management systems for inter-organizational workflows has been the need for manual redefinition of coordination points in the process models when either organization changes its processes. Coordinating communications between production-chain organizations are usually based on the notification of completion of tasks by one party which constitute pre-conditions for activity in another organization; autonomous task changes disrupt coordination. This paper describes a workflow re-coordination model and corresponding expert support system based on workflow goals which are more stable than the low-level machine, role and technique dependent activities which implement them. Following development of the model, we describe a set of three elementary disruption cases which span a large number of common workflow changes. Using these cases we demonstrate that common coordination disruption situations can be totally or partially repaired by use of an expert coordination subsystem.

Keywords: Workflow management system; Coordination; Goal-based process modeling

1. Introduction

Workflow management is the study and analysis of business processes with the explicit intent of interfacing, merging, expediting and/or redesigning those processes, when they are considered in their full context. Thus, workflow management differs from traditional industrial engineering process studies primarily in scope. Workflow management *intends* to cross divisional and traditional single-output-process boundaries in the search for global (organizational) process effectiveness. Workflow management includes analysis, description, design, and augmentation of workflows (Georgakopoulos, Sheth, & Hornick, 1994). As the conception of the enterprise becomes global and virtual (Morita, Mukaigaito, & Hayami, 1996; O'Leary, Kuokka, & Plant, 1997) the challenges of semantic ambiguity and dynamic task change and substitution are introduced into inter-organizational workflow management by the necessary *autonomy* of cooperating organizations (Vaishnavi & Kuechler, 2005).

Most existing WFMS are based on 'manufacturing models' of work where precise specification of activities and their execution times are (presumed) critical. Yet many work processes, especially knowledge (office) work, necessarily incorporate slack time to handle indeterminacy and frequently specify desired results *intentionally*, that is the process goal(s) are significant, but the details of execution are not (Davenport, Jarvenpaa, & Beers, 1996; Hewitt, 1991). As organizations become more virtual and an increasing number of non-strategic functions are subcontracted, traditional production workflows take on the same character of being more readily specified by intention than by concrete task.

Fig. 1 shows a macro view of the business environment. Processes in any organization (X, Y or Z) are dependent on

---

☆ This work is partially supported by NSF Research Grants, IIS-9810901 and IIS-9811248, and a research grant to the second author from the Robinson College of Business, Georgia State University.

[*] Corresponding author. Tel.: +1 775 784 6910.
E-mail addresses: kuechler@unr.edu (W.L. Kuechler Jr.), vvaishna@gsu.edu (V.K. Vaishnavi).

0957-4174/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.09.014

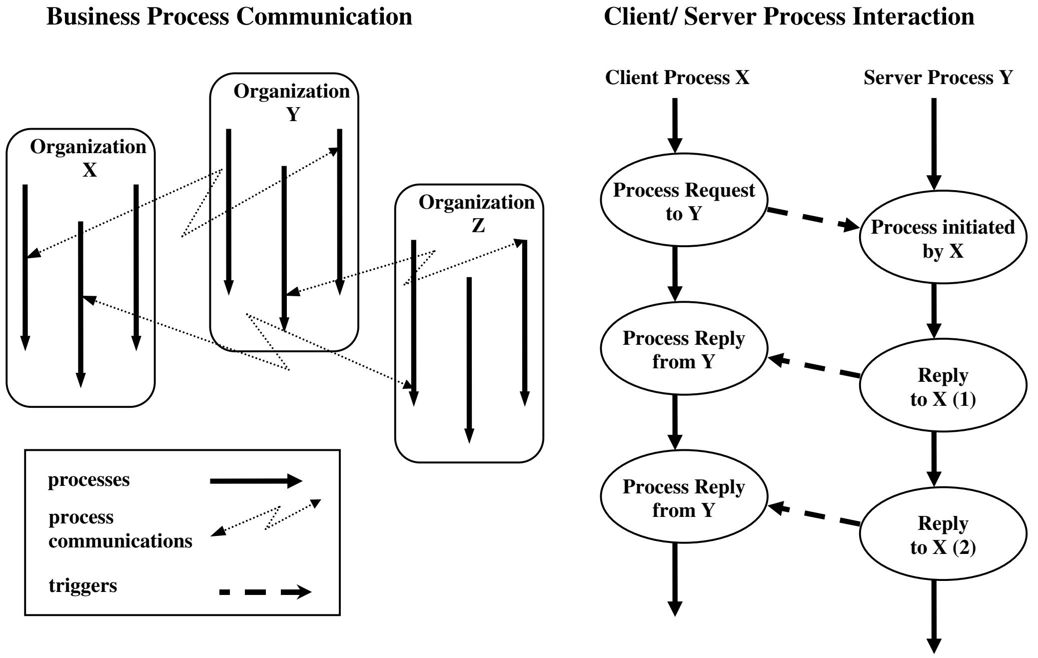


Fig. 1. Cooperation through Process interaction (adapted from Morita et al., 1996).

information from concurrent processes in their trading partners. The client–server portion of the diagram illustrates this dependency. Though the process interactions diagrammed in Fig. 1 may be manual or automated, our research is concerned with situations where the processes are controlled by automated workflow management systems (WFMS), and the process communications are between interoperating WFMS. Each WFMS operates from a model of a work process. A *trigger* (Joosten, 1994) in a WFMS signals the satisfaction of a set of preconditions that cause the enactment engine to perform (or schedule or assign resources to) a pending activity in a sequential process.

Recent work by the process modeling and workflow management communities defines the coordination of tasks across semantically heterogeneous environments as a significant problem area (Andersson, Bider, Johannesson, & Perjons, 2005; Buhler & Vidal, 2005; Khomyakov & Bider, 2001). We directly address some of the semantic issues in task interpretation that have been identified by these authors and others, for example (Agostini & De Michelis, 2000; Klein, Dellarocas, & Bernstein, 2000; van der Aalst & Jablonski, 2000), as *not* adequately addressed by prior WFMS and process research.

The problem of coordination disruption through autonomous activity changes in work processes is illustrated in Fig. 2. The figure shows a highly simplified definition of a security clearance process (Vaishnavi, Joosten, & Kuechler, 1996) in which a trigger, *preliminary_clearance_complete* in a process is described in terms of the completion of activity: *check for in state criminal record*. The trigger results in training for a new hire in parallel with the conclusion of the security clearance and having the training begin prior to a complete clearance saves significant time. When the process is redefined, as would likely be the case if the process were shifted to a subcontracting site, the trigger no longer exists

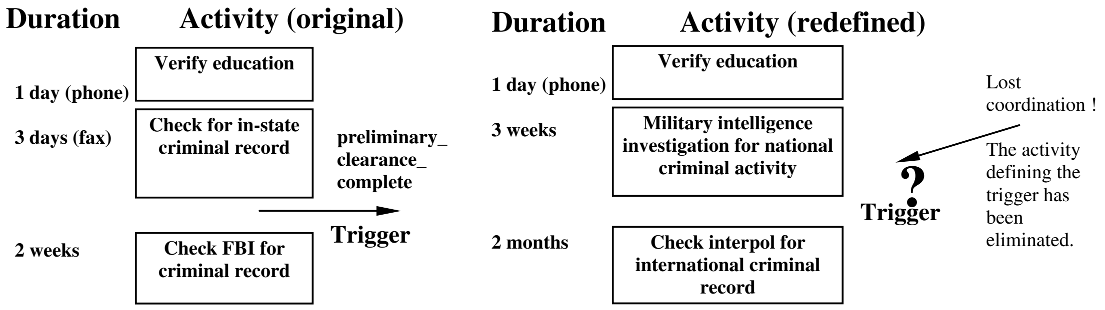


Fig. 2. WFMS trigger defined as process-state.

and training is delayed until coordination can be reestablished between the contractor and subcontractor.

Traditional work process modeling methods such as SADT and trigger modeling (Joosten, 1994) assume a deterministic, fixed element model (Lei & Singh, 1997; Yu, 1995). The system is closed, that is, the activities that make up the various processes are predefined. As indicated above, this assumption is almost certain to be violated when autonomous WFMS seek to interoperate. Autonomous groups, especially in outsourcing situations, must be free to change process details. Thus we seek some more robust method of specifying trigger conditions than matching to fixed patterns of activities.

Our prior research in this area (Vaishnavi & Kuechler, 2005; Kuechler, Vaishnavi, & Kuechler, 2001; Vaishnavi et al., 1996) has addressed the issues at a conceptual level and has developed formal models of the contractor–subcontractor relationship based on the workflow management consortium (WFMC) workflow enactment model. The research reported in this paper builds on that research to articulate a computable goal-based coordination model to implement the formal models and develops the computable model into a working prototype.

In the next section of the paper (Section 2) we survey the work of other researchers to introduce flexibility into process definitions. Section 3 develops the goal-based workflow coordination model and Section 4 presents the architecture of the corresponding expert system. Section 5 discusses the expert system implementation, introduces three exemplars of coordination disruption as test cases and describes the system response to these cases. In a concluding section we discuss some of the ways in which the system and model can be augmented, and our on-going research with the technique.

2. Search for flexibility in workflow and process modeling

In our research on coordinating inter-organizational WFMS, we have surveyed literature from multiple fields for approaches to similar problems and alternative approaches to coordinating complex systems. Coordination problems are ubiquitous, and in all fields, we find the recognition of the inadequacy of fixed representations for modeling real-world work situations.

Cao and Sanderson (1995) find fixed representations of process inadequate even for the constrained field of robotic workcells. Traditional deterministic planning representations are simply not robust enough under real-world conditions. Their approach is to expand traditional Petri net models of process by allowing the transitions between states to be expressed as fuzzy variables, interpreted by production rules. While suggesting interesting techniques, their research addresses a sufficiently different problem that their solution is not immediately applicable to workflows. Specifically, their domain requires fixed sequencing of activities, and many workflows, as noted in the literature, are dynamically resequenced in response to changes in the environment.

The HFBP model (Morita et al., 1996) of business workflows is specifically intended to describe work processes between interoperating workflows in autonomous organizations. Drawing from research in fault tolerant software systems, they model processes in terms of an attribute grammar. While successful in addressing certain types of error recovery in interoperating WFMS (due to inherent capabilities of attribute grammars) they note that the more general problem of dynamic change in process definitions for interoperating workflows is not sufficiently handled by the attribute grammar approach alone.

Much research in office information systems (OIS) describes the complexity of actual work environments, and is acutely aware of the inadequacy of fixed process models for support of real-world work situations (Boland, Maheshwari, Te'eni, Schwartz, & Tenkasi, 1992; Gerson & Star, 1986; Hewitt, 1986, 1991). The following five models come from that area:

(i) The partially shared view scheme (Lee & Malone, 1990) is a technique for enhancing the utility of computer supported collaborative work systems used between autonomous groups with differing world views and correspondingly different functional requirements. The scheme has been implemented in the Object Lens collaborative work support system, and attaches a type hierarchy to each work object the system supports. The type hierarchy is assumed at least partially shared (especially at its higher levels) by all users of the system. By determining correspondences with a known hierarchy, an object otherwise unknown to a user group can be at least partially recognized and usefully processed. Our workflow model adopts this technique by analogy: the knowledge in our hierarchy is different, but our partial matching to a hierarchical structure to make inferences about an unknown entity is very similar.

(ii) The AMS formalism (Ang & Hong, 1994) is both a model and a collaborative work support system implementation. The formalism can be characterized as "loosely grammatical" in that the activity sequences that make up a workflow are sequentially expanded from a root "abstract activity" by a series of productions. Stressing flexibility, however, it does not attempt to formalize the grammar (and obtain the benefits of following a well researched formalism) in the manner of HFBP. The allowed productions by which non-atomic activities can be expanded are (1) an elementary activity, (2) an activity network, and (3) a *memory organization packet for activities* (MOPA). The MOPA draws from the work of Schank and Abelson (1977, 1995) and others in natural language understanding and is responsible for much of the flexibility of the system. A greatly simplified description of a MOPA is as an abstract *script* that represents an activity sequence in general terms, and is instantiated to actual activities by the run-time context. Though more accommodating to actual work situations than the Petri net models it augments, AMS is essentially a single system model, and has no mechanisms for preserving coordination between interoperating systems which make use of its flexibility.

(iii) The Promanand system (Karbe, Ramsperger, & Weiss, 1990) is an office support system using a "circulating folder" metaphor. It is unique in that it is the most determined attempt to date to support a deterministic view of office work through the exhaustive enumeration of exceptions conditions for a given work environment. The lack of success of the system is viewed by Ellis and Wainer (1994) and others as support for the position that any non-adaptive system will ultimately prove inadequate to real-world work environments. Further support comes from Hewitt (1991) who has characterized office systems as *open systems* which cannot be analyzed a priori.

(iv) Mahling and King (1999) point out that nearly all of the problems of inflexible computer assisted workflows were first encountered by researchers in office information systems (OIS) in the 1980s. They recognize that workflow models based on goals are more flexible and robust than task descriptions, and their PolyFlow system dynamically expands goal-based process definitions into activity plans for the extremely demanding office work environment. Their approach coordinates changed activity sequences for different agents in a distributed workflow through automated support of human–human negotiation. While acceptable in the single non-repeating project office environment they envision, a wholly automated re-coordination approach seems desirable for repetitive, production workflows.

(v) Similar in many respects to PolyFlow is the case-handling approach of van der Aalst, Weske, and Grunbauer (2005). Directed also at office workflows, the distinguishing feature of case-handling, is that it is data driven rather than process driven; activities are enabled based on data availability rather than activity completion preconditions. While allowing for very flexible sequencing of activities, case-handling systems require continuous manual attention and extreme visibility of activity and context data as is unlikely to occur in a contractor–subcontractor production workflow.

Buhler and Vidal (2005) propose that adaptive workflow enactment will, based on historic and functional considerations, ultimately be enacted through agent-based systems. In this vision a form of high-level business process description language will be used to compose ensembles of agents into an "initial social order" for enactment of the workflow. As these software agents negotiate with each other and continuously search for web-based alternative services, workflows will be dynamically responsive to environmental changes throughout their enactment.

AgentWork (Muller, Greiner, & Rahm, 2004) is a WFMS that seeks to adapt to workflow exception conditions by specifying high-level logical rules for approaches for rescheduling activities. The system evolved in a medical environment where timely treatment is vital and so this system focuses on the detection of temporal exceptions and even projects time overruns before they occur. When exceptions are detected, a rule-driven scheduling system reschedules activities and may introduce substitute activities with different execution times or treatment applicability.

We note that while the workflow enactment agents of Buhler and Vidal (2005) as well as the rules of the AgentWork system (Muller et al., 2004) implicitly use goals as the higher level ordering principle for adaptive workflows,

Table 1
Flexible workflow models

Model	Authors	Term	Primary flexibility enhancement
Fuzzy Petri nets	Cao and Sanderson (1995)	Short	Fuzzy transitions to augment Petri net models
HSBP	Morita et al. (1996)	Short	Attribute grammar description of process
PSV	Lee and Malone (1990)	Short	Documents carry type/behavior hierarchy allowing partial interpretation
AMS	Ang and Hong (1994)	Both	Generalized scripts, which are situationally instantiated
Prominand	Karbe et al. (1990)	Long	Exhaustive enumeration (attempted) of all exception conditions for given environment
PolyFlow	Mahling and King (1999)	Long	Goal-described processes dynamically expanded to executable task sequences; negotiation support for exceptions
AgentWork	Muller et al. (2004)	Long	Rule-based rescheduling or substitution of activities when workflow exceptions (especially temporal) are detected
Multiagent workflow enactment	Buhler and Vidal (2005)	Long	Workflows are initiated by composing ensembles of software agents with a business process definition language. The agents then dynamically direct the enactment of workflow activity sequences
Case-handling	van der Aalst et al. (2005)	Both	Activities are enabled based on data availability rather than activity completion preconditions

neither system explicitly mentions goals. However, actions of the software agents are guided by goals in their search for alternative activity sequences and the rules that guide AgentWork have of necessity been a priori derived from the higher level goals of the processes they enact.

Table 1 summarizes the adaptive/flexible work process models discussed in this section. Soffer (2005) distinguishes between two forms of workflow flexibility: *long term* adaptability sometimes referred to as *agility* in processes and *short term* flexibility which is the ability to recover from small, short term changes in a way of working. We have categorized the models in Table 1 according to this criteria for comparison to our model, which is specifically aimed at dynamic recovery from small workflow changes (*short term*).

While all the research overviewed in this section is motivated by the understanding that actual work processes are highly situational and require adaptive models, none except HSBP considers the distinct, but closely related problems inherent in modeling *multiple, tightly-coupled autonomous production* WFMS. HSBP outlines a grammatical model for single systems, but specifically reserves the modeling of interacting systems for later research. In the next section we demonstrate a reconceptualization of workflow coordination that allows both process flexibility and tight coupling between trading partner WFMS.

3. Goal-based coordination model framework

Enhancing conventional activity-based process models with goal (intentional) information has been suggested as a basis for introducing greater flexibility into automated process enactment since at least the early 1980s (Mahling & King, 1999). In analyses of workflow meta-models (Lei & Singh, 1997), it is suggested that no other technique can provide the directive information required to dynamically re-sequence tasks in response to error conditions. However few models or WFMS have been developed that actually operationalize and use goal information.

Our workflow coordination model captures goal information about processes in a formal structure and uses the information straightforwardly to reason about changes to process activity sequences. We have named our model after its most distinguishing feature: a hierarchical overlay of process intentions (HOPI). Unlike agent-based approaches which require dedicated workflow systems (cf. Mahling & King, 1999), HOPI works as an overlay to augment most existing workflow management systems.

Several HOPI attributes are unique in workflow and process modeling: first, the goal hierarchy is used not for planning, replanning or evaluating a process but rather for *recognition of similarity* between processes; second, coordinating communications between cooperating workflow actors are elevated to the status of first order activities rather than pre or post-condition methods within an activity object. As such they are not bound to any specific activity, but like all activities within HOPI, are linked instead to the goal hierarchy. Indeed, unique to HOPI, coordinating communications are *assigned goals independently of activities* so that they can be rescheduled independently of activities.

Fig. 3 shows a high-level object model of the same process modeled in Fig. 2. Note that the trigger, *preliminary_clearance_complete* is an object at the same semantic level as the other activities in the process. Subject to semantic constraints derived from process goals in addition to timing and other hard constraints, it can be rescheduled and replanned to preserve coordination even under process change or activity substitution.

The primary data structure of the HOPI model is a *work definition* (WD) that specifies a hierarchy of intentions (goals) for the activities of a work process. The WD is an *overlay* to the activity sequence of the workflow and thus is generally applicable to any process representation and to the many WFMS currently in use.

Fig. 4 illustrates the definition of the WD used by the model. The *root goal* is the overall purpose of the work, and the name by which the work is commonly known in the organization. The layer of *subgoals* is generated during process design or specification, and represents a process of stepwise decomposition of the root goal into its components (Simon, 1977). When the subgoals are well enough defined to be implemented, they are linked to the *generalized functionality* by which they will be enacted. Incorporating the concept of functions into the WD more closely models the manner in which workers conceive their

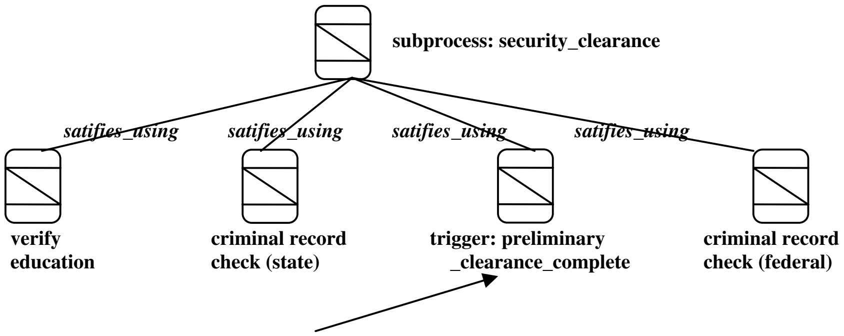


Fig. 3. WFMS trigger reconceptualized as a semantically defined, independent communications event.

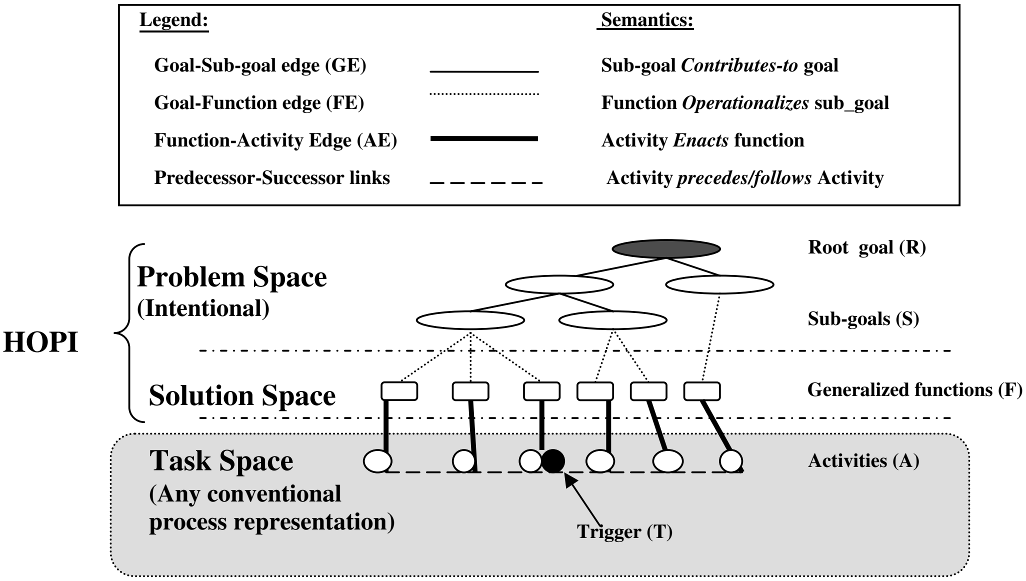


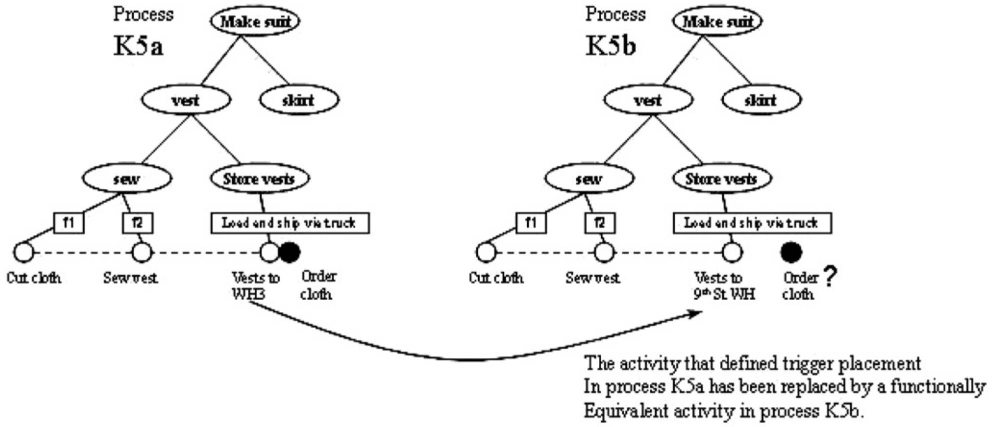
Fig. 4. A hierarchical structure of intentions as an overlay of process representations.

activities than use of goals alone (Schank & Abelson, 1995). The terminal (leaf) nodes of the WD are the actual work *activities* containing predecessor/successor information.

Fig. 5 shows the work definition instantiated for one of our three exemplar test cases: a simple activity substitution in the manufacture of a woman's suit, consisting of a vest and a skirt. Production of the vest is decomposed into *sewing* the vests and *storing* the vests until final packaging. The function that implements the storage subgoal is *load&shipViaTruck*, which is concretely realized by the activity *vestsToWH3.* The interprocess *trigger*, a communication to a trading partner to order cloth for other manufacturing processes (not shown) is linked to the completion of *vestsToWH3*. The linkage of a coordinating communication to a specific predefined activity state is common in WFMS and renders systems brittle under change. Following work redefinition (K5a → K5b in Fig. 5), the concrete activity has been changed to *VestsTo9thStWH*, resulting in coordination disruption and the failure to order the cloth in a timely manner.

The dynamic workflow re-coordination model depends on similarity inferences between an original process definition and an altered definition, both modeled with HOPI.

Fig. 6 illustrates the similarity concepts used to interpret the intentionally specified work definitions. The WD is a tree structure in which each element is a *node*. The path from the *root goal* to any activity node is termed the *intentional context* (IC) of the node, since tracing the path yields the semantics of the node. When ICs of activity nodes in different WDs are compared beginning at the root, the lowest identical node of the ICs is called the *lowest common point* (LCP) of the ICs. For example, if nodes 1 through 4 of WD's A and B in Fig. 6 are identical, but node 5 differs in A and B then node 4 is the LCP of the ICs. The depth of the LCP is a measure of IC similarity. Activity nodes also



The activity that defined trigger placement
In process K5a has been replaced by a functionally
Equivalent activity in process K5b.

Fig. 5. Original and altered work definitions for test case.

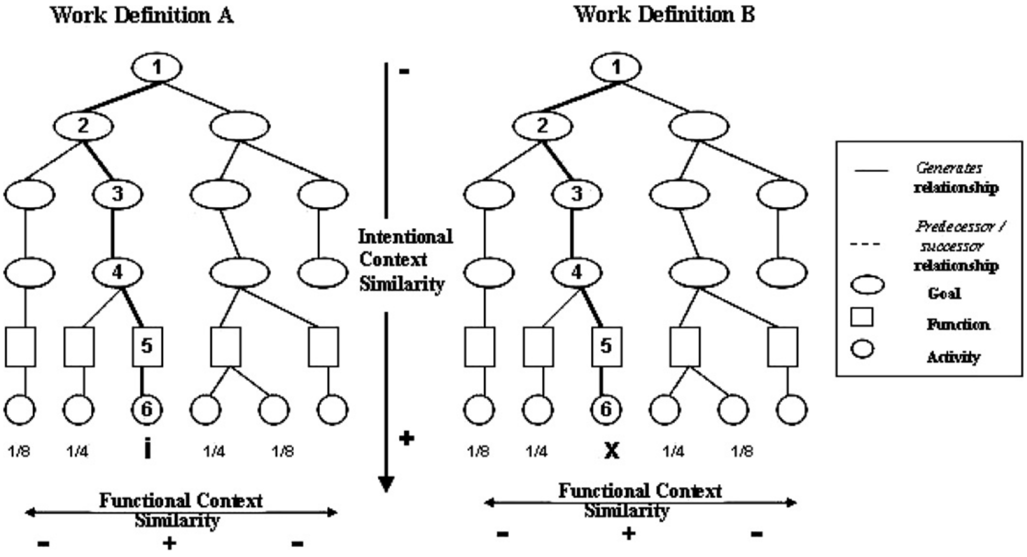


Fig. 6. Similarity inferences based on the work definition.

have a *functional context* (FC), which is loosely defined as the *set of activities adjacent to a given activity node*. Functional context similarity between activity nodes **i** and **x** in two WDs is measured by an index ranging from 0 (no similarity) to 1 (identical). The index is computed by adding a weight to the index for each activity node position relative to **i** and **x** for which the nodes are identically defined. Weights diminish geometrically with distance from **i** and **x**, as shown in Fig. 6. When information on the expected duration of the activities is available, a further measure of functional context similarity is *functional context duration*, the ratio of the times from the start of the process to the trigger activity and from the trigger activity to the end of the process in the original and the altered process expressed as a percentage.

Using the example of Fig. 5 to illustrate the use of the similarity measures, note that both activity ⟨vests to WH3⟩ in process K5a and activity ⟨vests to 9th StWH⟩ in process K5b have identical *intentional contexts*; the path down the intentional tree from the root goal ⟨makeSuit⟩ is the same in both cases. Further, the activities that precede and follow activity ⟨vests to WH3⟩ in process K5a are identical to those that precede and follow activity ⟨vests to 9th StWH⟩ in process K5b and those activities in both processes implement the same function ⟨load&shipViaTruck⟩. Thus the *functional contexts* are also identical. Together these measures strongly indicate that the change to the workflow activity is nominal and the trigger may be attached to the new activity with no loss of information. This is precisely the determination of the expert system as shown later in Fig. 13.

The similarity heuristics are based on Barsalou's (1983) work on ad hoc categories and the work of Schank and Abelson (1977, 1995) on the relation of scripts and goals to understanding. The key concept is that when activities and their goals are structured in a conceptual hierarchy as in our WDs, each higher level node serves as a cognitively organizing principle for the nodes beneath it, enabling intensional classification and recognition of lower level nodes.

In the *state-oriented view* of processes (Khomyakov & Bider, 2001) a business process is a trajectory in a multidimensional state space where each change of state (motion in the state space) takes the process closer to one or more goal states. Under this view, two processes are considered *similar* if (1) there is an isomorphic mapping from one state space to another (2) the process spaces have similar goal states (3) the processes have the same valid movements in the state space toward the goal (Andersson et al., 2005). Although one is derived from a cognitive model of process similarity perception and the other from a state-space formalism, HOPI and the Khomyakov–Bider process views are mutually reinforcing; our re-coordination model operationalizes all three Khomyakov–Bider similarity metrics.

Coordinating through goals rather than specific activities is closely analogous to the software engineering procedures by which subroutines are made loosely coupled (less dependent) in order that the system as a whole should be more robust (less fragile).

Prior to discussion of the operation of these principles within the expert system, a brief discussion of the usage scenario envisioned for the model will provide a high-level context (see Fig. 7). Two business entities, a contractor who requests work and a subcontractor who provides the work are involved in each use of the model. Both entities manage their processes with compatible WFMS and at some initial point share a common process definition for the subcontracted work. At a later time the subcontractor unilaterally changes the definition of the work being requested by the contractor; however the contractor requires notice of some intermediate point in the process (a trigger) to enable scheduling efficiencies. The coordination model attempts to determine the appropriate point

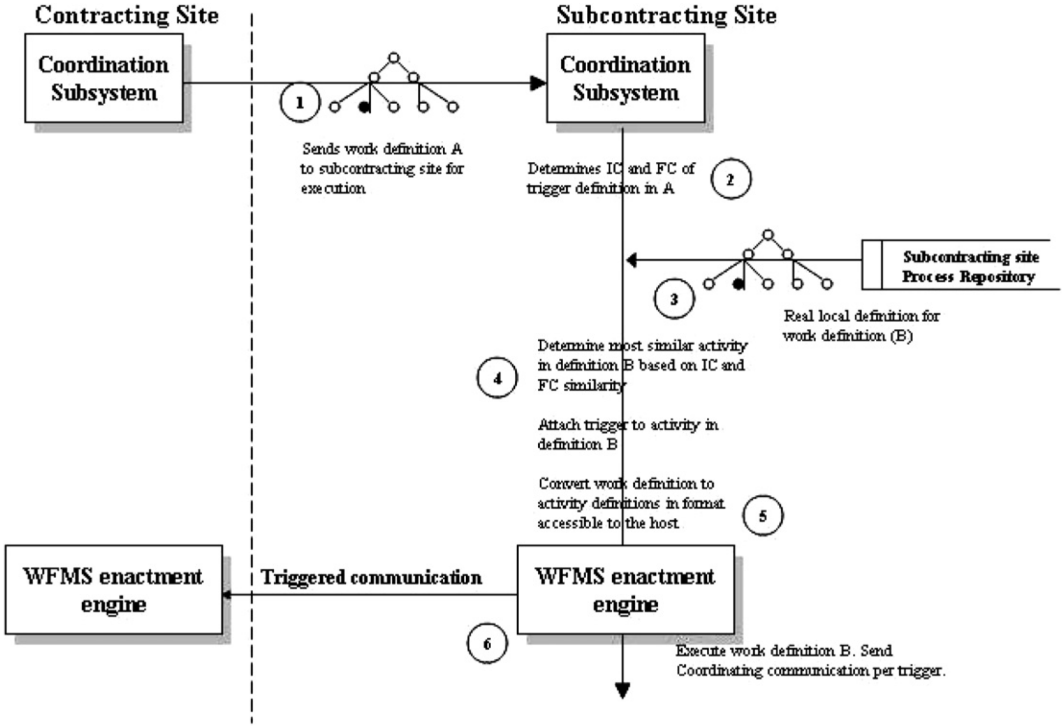


Fig. 7. A usage model for the coordination subsystem.

in the altered definition to schedule the trigger, eliminating the user communication and reprogramming that would otherwise be required.

This usage model corresponds to a *reference process-based multi-enterprise process* (Schuster, Georgakopoulos, Cichocki, & Baker, 2000) in which cooperating partners construct relatively long-lived virtual enterprises. Also referred to as *outsourced workflows* (Schulz & Orlowska, 2004), such inter-organizational processes are normally tightly coupled through work state dependencies which limit flexibility. However, the HOPI subsystem enables autonomous process redefinition, an adaptiveness normally found only in service-based workflows while retaining close, state-based coupling.

4. An architectural view of the coordination subsystem

The subsystem is divided into two major components, a *process architecture* and a *knowledge architecture*. This division is typical of knowledge engineering, where a *Process* and an *Expertise* analysis model are defined for every system (Chandrasekaran, 1986; Giarrantano & Riley, 2005; Tansley & Hayball, 1993). The *knowledge architecture*, the intelligence used in maintaining coordination, is an articulation of the HOPI intentional description of the workflow rationale as described in Section 3. The *process architecture*, which illustrates the relationship of the major functional modules required for dynamically maintaining coordination between loosely coupled (autonomous) workflows, is shown in Fig. 8. The basic structure is similar to well known architectures for analogous tasks: planning, understanding from plans, and constraint-based scheduling.

4.1. Process overview

Moving from left to right in Fig. 8: the original workflow definition (activities, their ordering, etc. as described in Section 3) and a modification of that original definition are available to the model. The comparison module determines differences and similarities between the definitions using the concepts of *functional context* and *intentional context* as set out in the previous section. Note that comparison is restricted to those portions of a *Work Definition* that contain triggers. The model does not concern itself with changes in any portions of a process that do not constitute *conceptual* or *functional* contexts for triggers.

After comparison, *difference data* is passed to the interpretation module. The interpretation module attempts to 'make sense' of the differences, and reconcile them to known elements in the original process definition. If this activity is successful, as judged by evaluation heuristics, the 'recognized' activity set is passed to the scheduler. That module attempts to reschedule coordinating triggers in a manner that satisfies both 'hard' constraints, such as fixed times for events, and fixed predecessor/successor relations and the semantics attached to the coordinating triggers.

The *scheduler* applies domain specific and domain independent heuristics to the scheduling of coordinating communications. This is required, for example, when changes

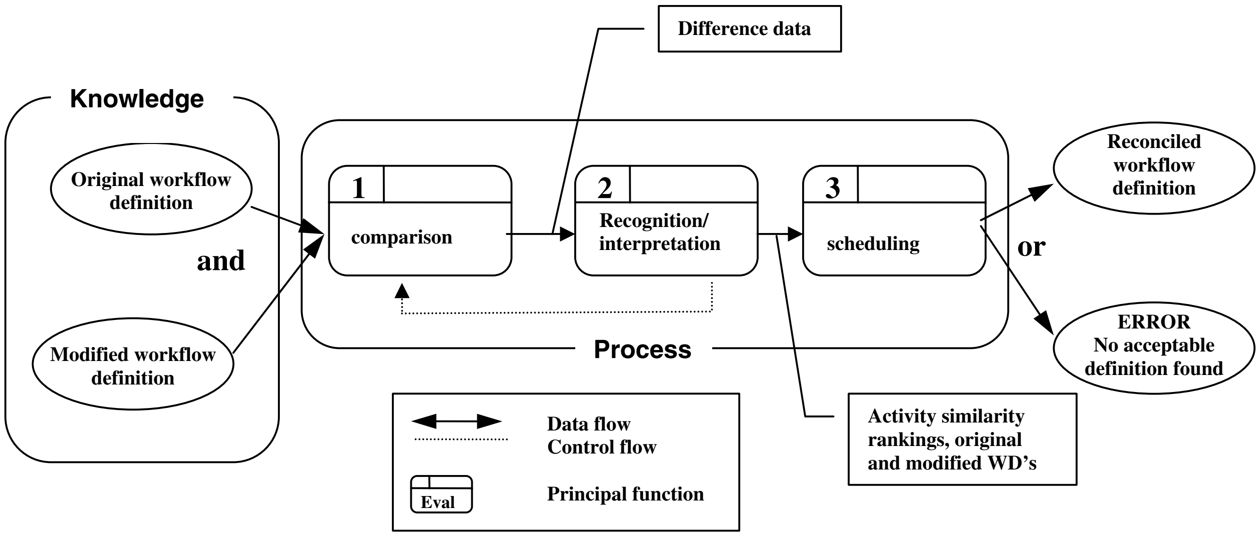


Fig. 8. Knowledge and process architecture for the dynamic WFMS coordination subsystem.

to the workflow definition have resulted in the elaboration of a single task in the original definition into multiple activities in the changed definition. Heuristics may be used to determine a point within the elaborated activity at which to communicate state completion to the modified workflow.

A process and a modification of it can always be reconciled at some level of abstraction. In general, very rough confidence factors can be assigned to a reconciliation (trigger rescheduling) based on measures of intentional and functional context. These factors along with the scheduling actions taken by the rescheduling algorithm are posted to a log file by the system for operator review.

5. Coordination system implementation

The coordination subsystem prototype has been implemented in CLIPS, the widely used NASA developed production system language (Riley, 2005). The HOPI workflow description hierarchy (described in Section 3) has been modeled as a tree structure of *deftemplates* nodes linked by predecessor/successor pointers. The architecture modules, *comparison, recognition/interpretation* and *scheduling* (described in Section 4) have been modeled by functions within different CLIPS modules (namespaces). We have chosen to use the CLIPS 6.23 Windows interface, even though it is primarily text-based since at this stage of the project only the developers are using the system.

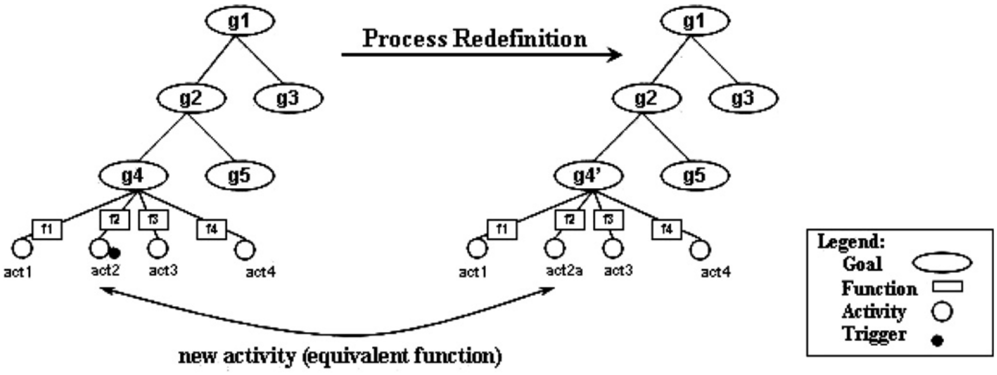
Each case – a set of CLIPS *deffacts* representing a HOPI model of two workflows, original and altered – is stored in a text file maintained with any text editor. In keeping with the usage model of Fig. 7, in the WD's the original workflow definition is termed *internal* (to the contracting company) and the altered definition in use at the subcontracting site is termed *external*. A partial listing of HOPI data structures definitions and data instances is shown in Table 2. Pseudo-code for one of the heuristic rules that reschedule triggers within the altered workflow definition is shown in Fig. 9.

Table 2
CLIPS data structures and instantiations for a HOPI work definition

HOPI node structure definition	Partial HOPI instantiation
(deftemplate goalNode (slot name) (slot parent) (slot workProcess))	(deffacts case1ExtGt "goal tree for external process, case 1X" (goalNode (name XverifyReferences) (parent xVerifyPersonalInfo) (workProcess XsecurityClearance) (goalNode…
(deftemplate activity (slot name) (slot parent) (slot scheduledDuration) (slot predecessor) (slot successor) (slot workProcess))	(deffacts case1Acts "activity sequence for case 1X" (activity (name XphoneFormerEmployers) (parent Xvalidate2) (scheduledDuration 48) (predecessor XfaxSchool) (successor XfaxToFBI) (workProcess XsecurityClearance) (activity…

RULE: *Simple activity substitution*
If activity linked to trigger in local WD is missing in external WD
AND
An activity in external WD (subAct) can be found with identical intentional context to triggering activity in local WD
AND
An activity in external WD (subAct) can be found with identical functional context to triggering activity in local WD
=>
Link trigger to subAct in external WD
Advise user of action and rationale
Transmit external WD to subcontracting site

Fig. 9. A trigger rescheduling heuristic.



General Examples:
(1) Upgraded equipment with identical functionality (transfer to Busselman operator (Bussleman is a lathe) vs. transfer to Whippet operator (Whippet is a new lathe)
(2) Change in physical plant (ingots transferred to building K vs. ingots transferred to 12th street plant
(3) Change in supplier

Fig. 10. Simple activity substitution (equivalent function, same level).

6. System run with test cases

6.1. Coordination disruption exemplars (test cases)

Any workflow activity *change*[1] of arbitrary complexity can be resolved into a repeated, sequential application of the elementary change patterns diagramed below in Figs. 10–12: (1) simple activity substitution involving a terminological change (identical function, different terminology), (2) multiple activities are subsumed in a new, more encompassing process step, (3) a single process step is articulated into multiple activities. The cases we describe are *patterns* as defined in Andersson et al. (2005) that provide the isomorphic mappings from an original process to an altered process.

6.2. System performance under test cases

Running the prototype with test cases is straightforward. The coordination subsystem code (HOPI.clp) is loaded into CLIPS using the (load) command. When run, the system asks for the name of a case file of facts defining an original and altered workflow as described above. Output from a run of the system with the elementary activity substitution example described in Section 1 (manufacture of a women's suit) is shown in Fig. 13 (cf. discussion of context similarity and intentional similarity in Section 3).

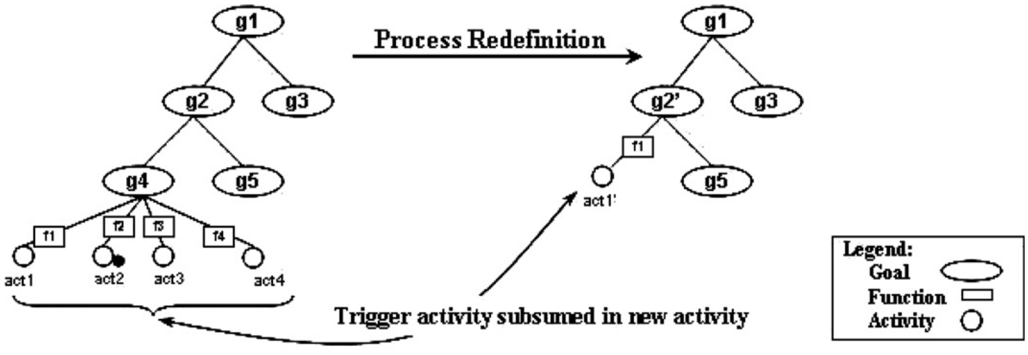
If attached to a working WFMS, the subsystem could transmit the new, reconciled workflow – the altered workflow with the trigger activity from the original workflow rescheduled within it – directly to the subcontractor's workflow enactment engine as described in the usage model of Fig. 7. Alternatively, the reconciled workflow could be transmitted to supervisory personnel for review prior to transmission to the workflow enactment engine.

7. Summary and future work

The functionality of the coordination subsystem in its present form is useful in many real-world situations since many alterations to work processes conform to our elementary change types. More complex changes typically evolve from the execution of multiple elementary changes over

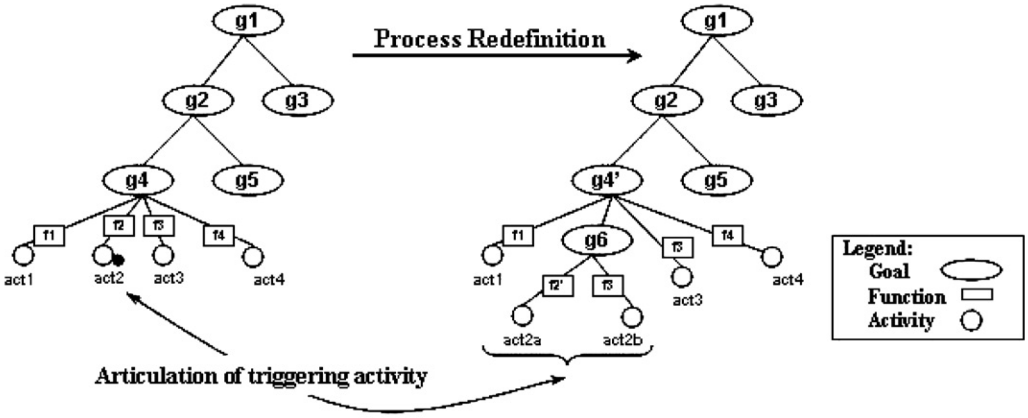
---

[1] Process redesign (substantial, deliberate activity resequencing) is not discussed in this paper nor addressed by the coordination subsystem in its current form.



General Examples:
(1) New equipment eliminates or combines multiple steps in a process
(2) Process has been subcontracted and information is no longer available
(3) New perception of **g2** subprocess goals
(4) Physical plant change or overall process reorganization

Fig. 11. New activity subsumes original activity (goal equivalent; level shifted up in the abstraction hierarchy).



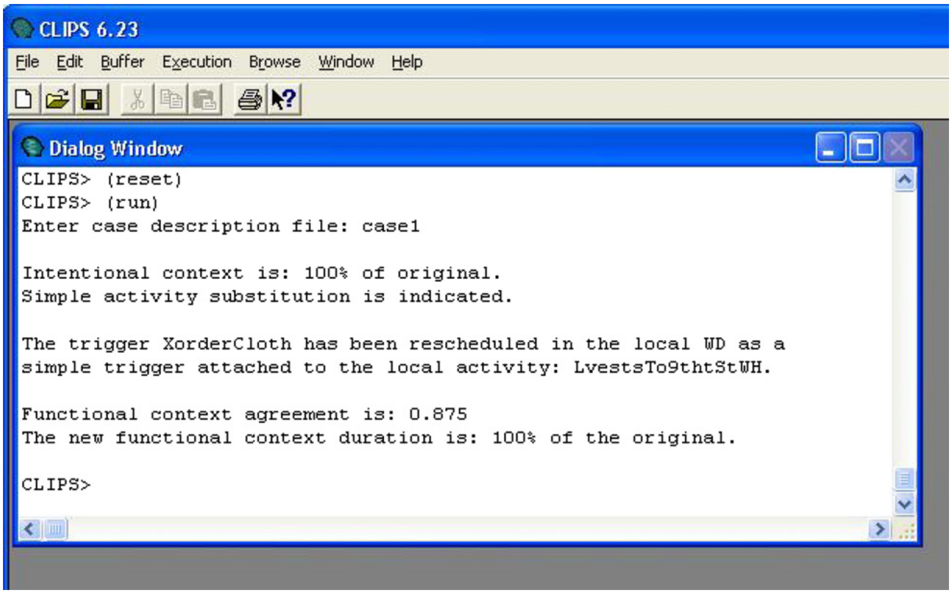
General Examples:
(1) Equipment substitution requires more process steps
(2) Process redefined at smaller granularity to gain additional control information
(3) Union or government regulations require safety or other new activities

Fig. 12. Articulation of original activity (goal equivalent; level shifted down in the abstraction hierarchy).

extended periods of time. Encouraged by the performance of the coordination model and prototype to date we have projects either planned or in process to extend this work in two directions.

First, we wish to validate and extend the heuristics used in the similarity computations and rescheduling rules of the model. We have already performed a pilot study to validate the instruments that we will use in a full concurrent verbal protocol exploration of expert process coordination problem solving. Analysis of the protocols should yield greater understanding of how experts reason when modifying coordinated inter-organizational processes. This experiment will validate the cognitive model on which HOPI is based.

A second research direction seeks to make HOPI and its associated re-coordination system more practical by partially automating the construction of the goal hierarchy through the machine analysis of textual documentation on a process. Most work processes in large organizations are accompanied by process manuals, design documents, memoranda to personnel involved in the process and other textual material. Using many of the same natural language processing (NLP) techniques that are currently used to monitor large bodies of e-mail and other textual material and for conceptual clustering of www inquiry results, we hope to construct tools for extracting process goals and functions from textual material ancillary to a work process.



```
CLIPS> (reset)
CLIPS> (run)
Enter case description file: case1

Intentional context is: 100% of original.
Simple activity substitution is indicated.

The trigger XorderCloth has been rescheduled in the local WD as a
simple trigger attached to the local activity: LvestsTo9thtStWH.

Functional context agreement is: 0.875
The new functional context duration is: 100% of the original.

CLIPS>
```

Fig. 13. Re-coordination system output.

The same NLP techniques can be used to automatically construct a lexicon (process ontology) that describes the process, its activities, its performers and their relationships. Using a lexicon, distributed artificial intelligence inference techniques (Vincenzo & Sessa, 2001) can be used to significantly extend the reasoning capabilities of the coordination subsystem. For example, a change from one functionally equivalent process step to another involving only a change of tool could be identified with much greater confidence using a lexicon. Inferences as to the similarity of new goals to existing goal/subgoal chains would be likewise enhanced.

An additional project which we hope to launch as soon as practical is to embed the coordination subsystem in an actual working inter-departmental workflow system proposed for the university of the first author. The information gained from real-world experience with the system would be impossible to gather any other way, and we look forward to the challenges this project holds.

References

Agostini, A., & De Michelis, G. (2000). In W. v. d. Aalst, J. Desel, & A. Oberweis (Eds.), *Improving flexibility of workflow management systems.. Business process management: Models, techniques and empirical studies* (1806 of LNCS, pp. 218–234). Berlin: Springer Verlag.

Andersson, B., Bider, I., Johannesson, P., & Perjons, E. (2005). Towards a formal definition of goal-oriented business processes patterns. *Business Process Management Journal, 11*(6), 1–14. Available fromhttp://www.ibissoft.se/publications/publications.htm.

Ang, J., & Hong, J. (1994). AMS formalism: an approach to office modeling and OIS development. *DataBase, 25*(4), 25–38.

Barsalou, L. W. (1983). Ad hoc categories. *Memory and Cognition, 11*, 211–217.

Boland, R., Maheshwari, A., Te'eni, D., Schwartz, D., & Tenkasi, R. (1992). Sharing perspectives in distributed decision making. In *Proceedings of CSCW '92*, November, 1992 (pp. 306–313).

Buhler, P., & Vidal, J. (2005). Towards adaptive workflow enactment using multiagent systems. *Information Technology and Management, 6*, 61–87.

Cao, T., & Sanderson, A. C. (1995). Task sequence planning using fuzzy Petri nets. *IEEE Transactions on Systems, Man and Cybernetics, 25*(5), 755–768.

Chandrasekaran, B. (1986). Generic tasks in knowledge based reasoning: high level building blocks for expert system design. *IEEE Expert, 3*, 23–30.

Davenport, T., Jarvenpaa, S., & Beers, M. (1996). Improving knowledge work processes. *Sloan Management Review*(Summer), 53–65.

Ellis, C., & Wainer, J. (1994). Goal-based models of collaboration. *Collaborative Computing, 1*, 61–86.

Georgakopoulos, D., Sheth, A., & Hornick, M. (1994). An overview of workflow management: from process modeling to workflow automation infrastructure. *International Journal of Distributed and Parallel Databases, 3*(2).

Gerson, E., & Star, S. (1986). Analyzing due process in the workplace. *ACM Transactions on Office Information Systems, 4*(3), 257–270.

Giarrantano, J., & Riley, G. (2005). Expert systems: principles and programming. Boston, MA: Thomson.

Hewitt, C. (1986). Offices are open systems. *ACM Transactions on Office Information Systems, 4*(3), 271–287.

Hewitt, C. (1991). Open information systems semantics for distributed artificial intelligence. *Artificial Intelligence., 47*(79), 79–106.

Joosten, S. (1994). Trigger modeling for workflow analysis. *Proceedings of CON '94 (Workflow Mgt. Div.)*. Vienna: R. Oldenburg.

Karbe, B., Ramsperger, N., & Weiss, P. (1990). Support of cooperative work by electronic circulation folders. In *Proceedings of the 1990 ACM COIS '90* (pp. 109–117).

Khomyakov, M., & Bider, I. (2001). Achieving workflow flexibility through taming the chaos. *Journal of Conceptual Modeling, 21*, 1–14.

Klein, M., Dellarocas, C., & Bernstein, A. (Eds.). (2000). Adaptive workflow systems [Special issue]. *Journal of Computer Supported Cooperative Work, 9*, 3–4.

Kuechler, W., Vaishnavi, V., & Kuechler, D. (2001). Supporting optimization of business to business e-commerce relationships. *Journal of Decision Support Systems,* (31), 363–377.

Lee, J., & Malone, T. (1990). Partially shared views: a scheme for communicating among groups that use different type hierarchies. *ACM Transactions on Information Systems, 8*(1), 1–26.

Lei, Y., & Singh, M. (1997). A comparison of workflow metamodels. In Stephen W. Liddle (Ed.), *Proceedings of the ER'97 workshop on behavioral models and design transformations: Issues and opportunities in conceptual modeling*. Available from http://osm7.cs.byu.edu/er97/workshop4/ls.html.

Mahling, D., & King, R. (1999). A goal-based workflow system for multiagent task coordination. *Journal of Organizational Computing and Electronic Commerce, 9*(1), 57–82.

Morita, M., Mukaigaito, T., & Hayami, H., (1996). HFBP: a hierarchal and functional business process model for describing distributed autonomous business environments. In *Proceedings of the NSF workshop on workflow and process automation in information systems, Athens, GA, UGA, 1996* (pp. 91–99).

Muller, R., Greiner, U., & Rahm, E. (2004). AgentWork: a workflow system supporting rule-based workflow adaptation. *Data and Knowledge Engineering, 51*, 223–256.

O'Leary, D., Kuokka, D., & Plant, R. (1997). Artificial intelligence and virtual organizations. *Communications of the ACM, 40*(1), 52–59.

Riley, G. (2005). *Clips reference manual: Volume 1 – Basic programming guide, June, 2005*. Available from http://www.ghg.net/clips/clips.html.

Schank, R., & Abelson, R. (1977). *Scripts, plans, goals and understanding: an inquiry into human knowledge*. Hillsdale, NJ: Lawrence Erlbaum.

Schank, R., & Abelson, R. (1995). *Knowledge and memory: the real story*. Hillsdale, NJ: Lawrence Erlbaum.

Schulz, K., & Orlowska, M. (2004). Facilitating cross-organizational workflows with a workflow view approach. *Data and Knowledge Engineering, 51*, 109–147.

Schuster, H., Georgakopoulos, D., Cichocki, A., & Baker, D. (2000). Modeling and Composing Service-Based and Reference Process-Based Multi-enterprise Processes. In B. Wangler & L. Bergman (Eds.). *Proceedings of advanced information systems engineering: 12th international conference, CAiSE 2000, Stockholm, Sweden, June 2000* (LNCS 1789/2000). Berlin: Springer Verlag.

Simon, H. (1977). *The new science of management decision* (Revised Edition). Englewood Cliffs, NJ: Prentice Hall.

Soffer, P. (2005). On the notion of flexibility in business processes. In *Proceedings of the 6th workshop on business process modeling, development and support, Porto, Portugal*. Available from http://lamswww.epfl.ch/conference/bpmds05/program/soffer_4.pdf.

Tansley, D., & Hayball, E. (1993). *Knowledge-based systems analysis and design: a KADS developers handbook*. New York: Prentice Hall.

Vaishnavi, V., Joosten, S., & Kuechler, W. (1996). Representing workflow management systems with smart objects. In *Proceedings of NSF workshop on workflow and process automation in information systems*, Athens GA.

Vaishnavi, V., & Kuechler, W. (2005). Universal enterprise integration: challenges of and approaches to web-enabled virtual organizations. *Information Technology and Management, 6*(1), 5–16.

van der Aalst, W., & Jablonski, S. (2000). Dealing with workflow change: identification of issues and solutions. *International Journal of Computer Systems, Science and Engineering, 15*(5), 267–276.

van der Aalst, W., Weske, M., & Grunbauer, D. (2005). Case handling: a new paradigm for business process support. *Data and Knowledge Engineering, 53*, 129–162.

Vincenzo, L., & Sessa, S. (Eds.). (2001). *Soft computing agents: new trends for designing autonomous systems*. Heidelberg [Germany], New York: Physica-Verlag.

Yu, E. (1995). Models for supporting the redesign of organizational work. In *Proceedings of the Conference on Organizational Computing Systems* (pp. 225–236). Milpitas, CA: ACM Press.