ACTIVITY CREDITING IN DISTRIBUTED WORKFLOW ENVIRONMENTS

Eric D. Browne, Michael Schrefl, James R. Warren
School of Computer & Information Science
University of South Australia

Keywords: workflow, adaptive, goal, activity, crediting, health, healthcare, care plans.

Abstract: Workflow Management Systems (W*f*MSs) are increasingly being introduced to deal with cooperative inter-organisational business processes. There are many situations in these distributed workflow environments, where, for a given business process, activities might be undertaken in one enterprise that overlap with, or repeat activities undertaken elsewhere. This paper examines such situations in the context of healthcare, where duplicated tests and procedures are costly and can have negative health impacts on patients undergoing unnecessary tests and interventions. Our approach is based on a two-tier goal/process representation of business processes and an execution model comprising a candidate discovery phase, followed by a component crediting phase. We introduce the notions of full vs. partial crediting, goal-level vs. activity-level crediting and examine the role that termporal constraints play in determining candidate components for crediting.

1 INTRODUCTION

Distributed workflow management systems (W*f*MSs), where an overall business process is undertaken by a number of different organisations, can lead to a situation where similar tasks might be undertaken by different organisations to achieve the same goal or part thereof. This duplication of services is inefficient, can lead to delays in achieving the overall goal, and can impact on the quality of the services being provided. This is particularly true in healthcare, where unnecessary interventions can have tragic consequences. Patients being managed for one or more chronic conditions, can be particularly prone to such adverse impacts of repeated interventions. A common example of this is with the prescription of medications, whereby several clinicians could be prescribing the same medication without being aware of similar prescriptions provided by other clinicians, either for the same, or sometimes for different reasons.

This paper introduces to workflow process modelling the concept of activity crediting, whereby an activity scheduled to be undertaken somewhere in the overall workflow, can be credited, either in part or in full, against a similar activity elsewhere in the workflow. Such crediting could be determined to some extent, at schema definition time, but quite often, with dynamically created workflows or subworkflows, the crediting may need to be undertaken at run-time, after the specific workflow instance has started. The semantics and rules for such crediting can be quite complex, with temporal aspects often playing an important role. Temporal deadlines and activity-outcome time to live parameters need to be taken into consideration when determining the potential for activity crediting to be viable. Another confounder discussed in this paper is the effect of intermediate tasks on activities that might be candidates for crediting.

There are also related but disimilar issues to activity crediting - those of activity discrediting, activity dependence, and activity conflict, some of which we also raise in this paper.

2 WORKFLOW in HEALTHCARE

In this section we look at the specific problem we are trying to address in healthcare, and how our research contributes towards a solution to this problem. We also introduce the notion of different *activity target objects*, which arise in healthcare workflows to a greater extent than in most other domains. For those readers unfamiliar with the domain of healthcare, it is important to appreciate that workflow in this domain,

D. Browne E., Schrefl M. and R. Warren J. (2004).
ACTIVITY CREDITING IN DISTRIBUTED WORKFLOW ENVIRONMENTS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 245-253
DOI: 10.5220/0002631702450253

is highly complex, varies substantially from case to case, is subject to many extraneous and unforeseeable inputs, and decisions are subject to incomplete and often imprecise data.

2.1 Problem

The motivations for the use of Workflow Management Systems in healthcare are twofold, firstly to improve health outcomes for individual patients, and secondly to make healthcare service provision more efficient. The primary focus of this paper is to address the duplication of services that often arise in healthcare service delivery. Such duplication affects both the quality of patient care, as well as the efficiency and cost of care delivery.

2.2 Requirements

Below, are outlined the primary requirements that need to be considered when addressing the above problem. Some additional issues related to these requirements, are mentioned at the conclusion to this paper.

2.2.1 Crediting Scenarios

Our requirements are based on a set of scenarios that enable crediting to be classified according to the following perspectives:
Full Crediting: Under some circumstances, the crediting of a goal or activity could be complete, in the sense that the goal or activity could be unconditionally dropped from the overall business process.
Partial Crediting: In many instances, it is unlikely that the completion of an activity somewhere in a workflow schema will make another activity completely redundant. However, the second activity may need to be modified to take into account the effects of the first activity. We refer to such compensations as "partial crediting". Partial crediting equates to constraints being placed on the crediting operations that modify the workflow schema at runtime. Such constraints restrict the validity of the crediting and could be based on time or on specific events.
Temporary Crediting: Temporary crediting refers to crediting being conditional upon the respective time at which similar activities occur. Depending upon their temporal dispositions, this could lead to one activity being cancelled entirely, postponed for some period, or foreshortened in duration.
Permanent Crediting: Sometimes, it will be appropriate for a goal, activity or data item to be dropped permanently from the workflow since its enactment or acquisition has been made redundant by a prior action.

2.2.2 Overlapping Tasks

It is important not to eliminate a target task, if its function overlaps that of the prior crediting or source task. For example, if a prior task achieves a blood pressure of 130/80 by, say medication, and a later task aims to achieve a blood pressure of 135/85 by exercise, it may be desirable not to credit the first task in order to eliminate the second, since there may be additional benefits gained by the exercise task. For the W*f*MS to be able to identify such overlaps, there must be sufficient detail expressed explicitly in the task definition, in this case as a post-condition of the exercise task.

2.2.3 Intervening Tasks

Even though an activity might have been identified as a candidate for crediting, it is possible that some intervening task, occuring after the first task, but prior to the second task in the crediting pair, could cause the crediting to no longer be valid.

2.2.4 Extraneous State Changes

Similar to the case of intervening tasks invalidating crediting as just described, we could have situations where unexpected state changes occur, to any of workflow state, patient state, environment state or resource state that similarly invalidate a possible activity credit. For example, a patient could be undergoing treatment for diabetes, whereby an exercise regime to reduce blood pressure credited and allowed for the skipping of a medication-based hypertension treatment. If the patient was physically incapacitated by some accident, then the exercise regime would no longer be a valid activity for treating hypertension, and the medication-based treatment might need to be reinstated into the workflow.

2.2.5 Unwanted Crediting

Unwanted crediting refers to situations whereby it is undesirable to cancel an activity because it has already been undertaken somewhere earlier in the workflow. For instance, a second opinion on a diagnosis may be an integral part of a workflow schema, in cases where it is important to have near certainty before proceeding down a path. Automatic crediting, and consequential cancelling of such second opinions would not be wanted or warranted.

2.2.6 Conflict Resolution

Akin to activity crediting is the notion of activity avoidance, due to potential conflict of outcome. Traditional W*f*MSs are unable to deal with such conflicts, since considerable domain knowledge is required to determine potential conflicts. Just like crediting, such conflicts could be resolved either at the goal level, or at the level of individual activities. Any facilities provided by an extended W*f*MS could be utilised to help resolve a conflict once it has been identified. E.g. users could be notified and workflow schemas adapted to help provide a solution. A W*f*MS could identify all actors in the healthcare team for a given patient care plan, and notify each actor through their appropriate communication channels. An additional activity could be automatically inserted into the current workflow schema to ensure that the conflict is resolved to the satisfaction of a nominated actor.

2.2.7 Quality Assurance

It is essential in any healthcare system to ensure that mistakes are minimized. Any form of activity crediting should be highly conservative, and subject to runtime validation and manual authorisation by approved and appropriate clinicians. There should be "break-glass" emergency exception handling to allow clinicians to overide perceived system-based activity crediting. Where crediting has already been approved by one clinician, such crediting should be made known to other relevant care providers for that patient.

2.3 Contributions

In this paper we describe mechanisms for eliminating redundant activities as well as partially crediting the work achieved by previous activities in a given workflow instance. Our contributions include the use of a two-tier methodology for representing and viewing business processes, based on separate goal and process views, and a two-phase methodology for firstly discovering, and secondly crediting selected components of the overall business process. We call the first phase *candidate discovery*, and the second phase *component crediting*. We identify two classes of crediting, namely *permanent crediting* and *temporary crediting*. We also describe a set of still unresolved issues that need to be addressed for activity crediting to be supported effectively in the clinical setting.

2.4 Approach Overview

The strategy we have adopted to address the problem of duplicated services hinges firstly on an approach to discovering candidate business process components for crediting and secondly on an approach to manage the crediting process itself. For candidate discovery, we utilise a two-tier methodology based on the separation of high-level goals from lower-level processes. For component crediting we adopt a self-modifying workflow architecture that embeds specific workflow modification activities into the workflow schema itself. These activities make use of dedicated crediting operators to achieve and manage the component crediting at runtime. In order to support these two aspects, we introduce a representational model and an execution model.

3 REPRESENTATIONAL MODEL

The representational model uses a two-tier goal/process architecture and a library and ontology of predefined task definitions as described below.

3.1 Two-tier Architecture

The two-tier representational model is based on defining a high-level goal view via a goal hierarchy (see 3.1.1), and a corresponding lower-level process view via a workflow schema (see 3.1.2). The workflow schema is derived from the goal hierarchy as described by (Browne et al., 2003). These two views provide for visually representing and interacting with the workflow for each instance.

3.1.1 Goals

Our approach builds on previous work (Browne et al., 2003), which introduces a two-tier model, based on the separation of high level *goals* from lower level *processes*. Although rarely addressed in the context of workflow systems, high level goals have been used extensively as a basis for Requirements Engineering specification (Dardenne et al., 1993; Jacobs and Holten, 1995; Gross and Yu, 2001; Mylopoulos et al., 1999) and have also been discussed extensively by the authors and proponents of the Asbru clinical guideline language, (Miksch et al., 1997) who couch goals in terms of plan *intentions*. Each activity in a workflow schema can be associated with the achievement of one or more high level goals. By decomposing goals into a goal hierarchy whereby the root goal corresponds to the overall objective of the health care for the patient, one can often identify and annunciate a range of ever-more specialised goals, culminating in goals which can be implemented by known specific best-practice activities. The goal hierarchy can then be mapped into a workflow schema for the patient, using a combination of clinical guidelines and organisational business rules and constraints. It is possible to identify potential candidates for activity crediting, even at the goal level.

We will use an example from Non-Insulin Dependent Diabetes Mellitus (NIDDM) management to illustrate a possible goal hierarchy, since diabetes management can involve many service providers (general practitioner, diabetes educator, nurse, endocrinologist, dietician, opthalmologist, podiatrist) and many potential activities. Consider the following goal hierarchy:-

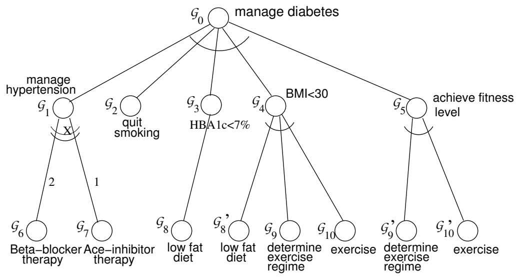

Figure 1: Goal View for Diabetes Management (simplified).

In fig.1 most goals are achieved by the achievement of all their immediate child goals, indicated by a single arc. Alternative goals are indicated by a double arc, as in goals $\mathcal{G}_6$(Beta-blocker therapy) and $\mathcal{G}_7$(ACE-inhibitor therapy). These two goals are mutually exclusive (annotated in the figure by an X), and are prioritised, such that ACE-inhibitor therapy is the preferred goal.

3.1.2 Processes

A *process* is a set of tasks or activities that achieves a goal. A process is implemented by a workflow engine that controls the activities, by either invoking specific activities itself, or placing them on the worklists of participants and flagging them as ready to be undertaken. The ordering of activities in a process is governed by a workflow schema. Thus, a process is an enactment of a workflow schema to achieve a specific goal. Workflows, and thus processes, may be nested and referred to as subworkflows or subprocesses. Each subprocess is designed to achieve a subgoal of the overall objective (root goal) of the goal hierarchy for that case. Such a process skeleton, derived from the diabetes management goal hierarchy is shown in fig 2, where each goal corresponding to its respective process is illustrated on the right-hand side at the corresponding vertical position in the diagram.

In moving from the goal view to the process view, we are adding domain, organizational and environment knowledge in order to elucidate the explicit set of activities, resources, conditions and control flow that needs to be applied. Every process $\mathcal{P}$, designed to achieve a goal $\mathcal{G}$, will have a set of activities A; a set of one or more objects upon which the process acts O (described below); a set of edges E that join activities, defining temporal dependence (flow) between activities; an associated set of resources R; achievement conditions G; and a goal validity time V. i.e.

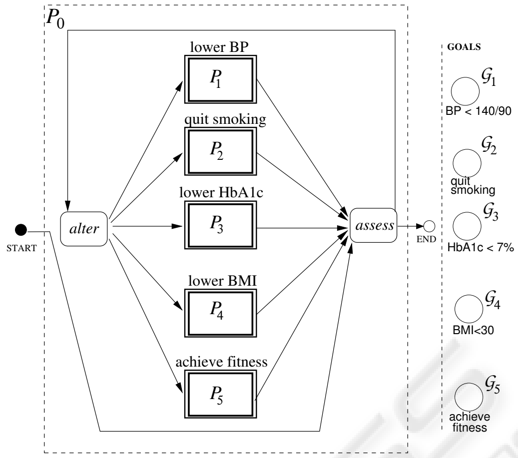

Figure 2: Process model for Diabetes Management (simplified).

$\mathcal{P}$ is described by the tuple (G, A, E, O, R, V). G expresses the boundary conditions on the goal, such as the achievement level, achievement tolerance, goal preference, goal priority, etc.

Fig.3 shows the expanded form of subworkflow P_4 of fig.2. This workflow illustrates the 3 processes corresponding to the subgoals needed to achieve a lower Body Mass Index(BMI). The first of these goals, *low-fat diet* is also a subgoal of G_3, lower-HbA1c.

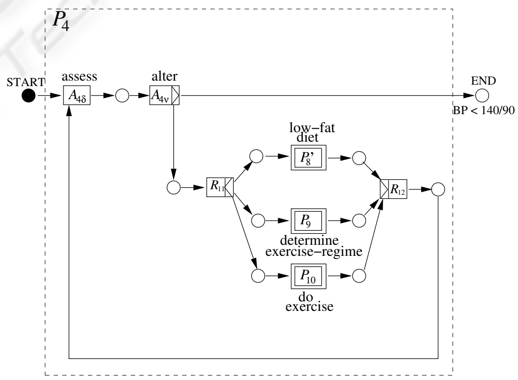

Figure 3: Process model for subworkflow P_4 - *LowerBMI.*

3.1.3 Task Library and Task Classification

In order for activity crediting to be viable, the W*f*MS needs to be aware of details of each activity in a given workflow schema. The best approach, based on goals as described previously, is for each schema to be derived from the goal hierarchy, as a set of nested subworkflows. Each subworkflow corresponding to a leaf goal is then constructed by assembling appropriate activities from a common organisational task library. This task library should be built from best-practice clinical guidelines (Field and Lohr, 1990), and should vary in scope according to the skills of the organisation. Tasks are classified according to an information model of healthcare concepts, so that tasks to accomplish similar goals are grouped/classified together. The intention, generic pre-conditions and generic post-conditions of each task are made explicit and stored in the task library. Candidates for activity crediting are then identified by the activity's path in the task hierarchy, the activity's intention, and the activity's data pre/post-conditions.

3.1.4 Activity Target Objects

In this section, we introduce the concept of an *activity target object* in order, firstly, to support those special activities in our self-modifying schemas that act on the workflow schema instance itself to perform crediting, and secondly, to support the necessary temporal constraints that may be required for crediting.

Workflow models are typically viewed as acting on a workflow case. Sometimes this case corresponds to a physical product, such as the production of an aircraft. Sometimes the case corresponds to a business service such as an insurance claim, travel booking or document processing. However, in the healthcare context, there are often a number of objects, or targets of a given activity or set of activities that need to be considered. At the highest level, we are concerned with providing a healthcare service. The particular service is often the major objective of the workflow schema. In this sense, we have two similar process targets:- the completion of the service, and the improved health of the patient that corresponds to this service case. The success of the former is often measured by the quality of service provided, independent of the patient outcome. The latter is measured by a change of state of the patient.

However, some parts of the workflow schema may act on secondary targets. For instance, a patient specimen, preparation of resources such as an operating theatre, analysis of patient data, analysis of the environment. An activity could be undertaken which teaches a carer how to care for a diabetic foot. Here the target might be regarded as a resource, or even part of the environment, since the carer may not be a participant in the usual sense of an initiator of a modelled activity, but merely the recipient of an activity undertaken by another service provider. Because of the flexibility required in the healthcare domain, we even introduce activities that act on the current workflow schema itself, in order to adapt the schema for changed conditions. A corresponding set of state parameters can be introduced to represent the state of these different targets at any one time. These are:-

- *workflow state*, S_W
- *patient state*, S_P
- *environment state*, S_E
- *resource state*, S_R

Workflow state S_W is further comprised of two components,

- *control state*, S_{Wc}, representing the workflow behaviour, and
- *data state*, S_{Wd}

The reason for explicitly defining *data state*, S_{Wd} is to clearly differentiate information that is a snapshot of real, continually changing, physical information from the physical information itself. For instance, a patient's blood pressure continually changes with time, whereas recorded blood pressure is a discrete set of zero or more values taken at specific instances of time. Theoretically, we can measure a patient's blood pressure at any future time t, but we may only have access to a small set of readings from the past, and may be unable to determine what the blood pressure was at any given moment in the past. Thus, a workflow data variable can be represented by a set of tuples, where any one $s \in S_{Wd}$ can be expressed as $s = (n, d, a, v_s, v_e)$, where n is the name of the state variable, d is the data value, a is the workflow activity that acquired or set this data value, v_s is the time at which data value became valid, and v_e is the time for which the data value is no longer valid. Thus $v_e - v_s$ represents the validity duration or time-to-live of the data value. Some temporal databases (Snodgrass and Ahn, 1985) only capture the start validity timestamp, assuming that updates to the data variable will automatically define the end of the previous validity time. In healthcare, we often have data being contributed from different sources whereby it is possible to have overlapping validity timestamps entering a shared repository. Many temporal databases also capture transaction time. In the healthcare context, we assume that all data is implicitely timestamped with its transaction time as would normally occur in patient Electronic Health Record (EHR) repositories. Some EHR-based systems may capture additional metadata regarding each data variable, such as the accuracy; the clinician's confidence in the value; a reference to the clinician who was responsible for this value of the data for a given activity.

Thus an activity target object O can be any one element of the set $\{C, P, W, E, R\}$ where C is the workflow case, P is the patient, W is the workflow schema, E is the environment, R is the set of resources involved in the case.

Healthcare activities are often categorised into *Investigations/Observations*, *Evaluations/Assessments* and *Interventions/Treatments*. From our definitions of state above, we can say that Investigations measure patient state (S_P) in order to change workflow data state (S_{Wd}); Assessments change S_{Wd}, often by temporal, spacial and other algorithms applied to prior values from S_{Wd}; Interventions are intended to change patient state (S_P).

4 EXECUTION MODEL

The two-phase execution model consists of a *candidate discovery* phase followed by a *component crediting* phase. These two phases are invoked at the commencement of the execution of each case, and whenever a change occurs to the status of any of the components identified by the *candidate discovery* phase as being involved in potential crediting.

4.1 Candidate Discovery Phase

Synergy refers to the similarity between two or more concepts. Workflow crediting aims at determining which elements of a business process are highly synergistic, and therefore candidates for crediting. We identify three types of components for which synergy can occur, namely those of goals, activities and data. A goal, activity or data variable has a high degree of synergy with another, if it contributes in a similar way to the overall objective of the business process. Taxonomies or classifications of concepts are important for the determination of synergy.

Before detailing the mechanism for candidate discovery, we first examine how goal discovery is undertaken using our goal-level representational model, whilst activity and data discovery are undertaken at the process-level.

Goal-level Matching refers to the identification of duplicate goals in a goal hierarchy for a specific healthcare service. Such goal hierarchies might be quite extensive in the treatment and/or management of chronic conditions, especially where comorbidities i.e. several concurrent conditions exist. Subprocesses (sets of activities) that would normally be enacted to achieve alternative or duplicated goals would be candidates for dropping or bypasssing in the overall workflow schema. Goal-level crediting allows localisation of modifications to specific areas of the workflow schema, minimising the overhead and side-effects of schema modification.

Process-level Matching refers to the identification of particular activities or redundant activity data items (state variables) that might be collected by different activities in the overall workflow. It is not sufficient to simply use the name of the activity as a means of identifying where such redundancies might occur. Many activities in healthcare are aimed at assessing or determining patient state variables, such as blood pressure, weight, blood lipid levels, etc. Many tasks could have the determination of these variables as either a direct or indirect target of the activity.

Candidate components for crediting are determined firstly by a goal-space search of the goal-hierarchy for matching goals. This is undertaken by following each path of the directed acyclic graph, from the root, to all leaves. Duplicate candidates are identified by the name of the goal and are further matched by their target achievement level. Temporal conditions are checked using the prescribed validity times associated with each goal. Once completed, if candidates are found, then these are placed on a candidate list for later processing. Mutually exclusive activities (alternatives) are checked, and if any in an alternative set has been commenced, then the remainder of the set is placed on the candidate list.

Next, the corresponding workflow schema is searched for synergistic activities. These are identified by the name and position in the task library of the template task used to create each activity. Any candidate activities are also placed on the list. Next, a search for data in activity post-conditions is undertaken to determine activities which collect or set identical data variables.

Finally, the candidate list is passed to the component crediting phase, which undertakes the crediting as described next.

4.2 Component Crediting Phase

Self-modifying Workflow: Activity crediting requires operators to support instance-level adaptation of workflow schemas, together with the cooperative interaction of healthcare participants. To support the required level of flexibility, we extend upon work (Browne et al., 2004) which introduces explicit schema modification tasks into workflow models of healthcare to ensure and facilitate the adaptation and modification of an abstract workflow schema at runtime for each patient case. Such activities are designed to change workflow state S_W (both S_{Wd} and S_{Wc}), resulting in additional activities, altered activities, replaced activities, deleted activities or altered flow. As such, these modification activities differ from conventional activities in that they act on their own workflow schema instance as the target object of the activity - hence the label, *self-modifying*. Considerable research, e.g. by (Ellis et al., 1995; Reichert and Dadam, 1998; Sadiq and Mangan, 2002; Manolescu and Johnson, 1999) and many others has formed the basis for much of the ideas leading to the concept of self-modifying workflow.

Every goal in the goal hierarchy has a corresponding workflow process in the workflow schema, such that the goal hierarchy maps to a nested set of subworkflows. Each subworkflow contains a goal assessment activity (*assess*) and a workflow alteration activity (*alter*). Component crediting is implemented by crediting operators, which are activity methods of the top-level *alter* activity (refer fig. 2), or in a similar *alter* activity in one of the subworkflows (e.g. within $P_1 ... P_5$ of the same figure). These *alter* activities have the workflow schema itself as the target object of the activity, rather than the patient, a resource, or the environment. A crediting workflow activity is invoked as part of each alter task and placed on the worklist of users who belong to the crediting role. Any candidate goals, activities or data variables are presented to the creditor for acceptance, if they correspond to the goal/subworkflow that is currently being executed. If the change is accepted by the creditor, then the corresponding crediting operator is used to apply the change to the running workflow schema, and to the goal hierarchy if appropriate.

Crediting operators refer to the set of workflow modification operators that play a part in any form of crediting. They are intended to alter the workflow schema, downstream of current activities, and do so by bypassing, adding, deleting, replacing or altering activities. Altering of activities may mean adding, deleting, replacing or changing data parameters of activities, specified as either pre- or post-conditions of the activity. We have identified the following workflow modification operators which specifically address changes to workflow schemas where one activity in the schema can be identified as providing some functionality similar to that of another activity in the same schema. These operators can be expressed as an ACTIVETFL (Müller, 2002) formula, where appropriate, utilising the temporal semantics available therein to conditionally apply a change to a schema, either by restricting the change to apply for some duration only, or in response to a specified event. Thus changes to the workflow can be specified as either permanent or temporary. Temporary operators, in turn, may need to credit, or uncredit a workflow component, and crediting can be done conditionally where the temporal constraint is known at the time of crediting, or else unconditionally, in which case a corresponding uncredit component is required. Temporal constraints are the most likely candidates to influence the viability of activity crediting. We need to tag certain state measurements and state changes with their validity time, or else have a mechanism by which the workflow engine can obtain this information generically from clinical knowledge. Pre-conditions for activities will often be specified in terms of predicates on state variables. E.g. "if patient blood pressure is greater than 150/90mm Hg, then ..". If the patient's blood pressure has already been measured by a previous activity, then not only does this need to be known, for activity crediting to be possible, but also, both the time of last recording, as well as the validity time, need to be known.

Here is a summary of the crediting operators supported:-

- *disableGoal*: disable a goal in the goal hierarchy, such that the corresponding workflow process is not undertaken. Goals are identified by their absolute path in the goal hierarchy, e.g.

  $$disableGoal(ManageDiabetes/BMIlt30/lowFatDiet)$$

- *enableGoal*: (re)enable a goal in the goal hierarchy. A goal may need to be reinstated if the crediting goal is not achieved, or if extraneous state changes cause a service provider to deem it desirable to reachieve the goal.
- *deleteGoal*: delete a goal from the goal hierarchy, such that the corresponding workflow process is removed and no longer undertaken.
- *skipActivity*: skip an activity from the current schema instance, for some or all of the remaining execution of the current case. The following rule:

  $$\begin{array}{l} WHEN\ completed(A_1) \\ THEN\ skipActivity(A_2) \\ \quad VALID{-}TIME[now,now{+}(7,day)], \end{array}$$

  states that activity A_2 should be skipped for 7 days on the completion of activity A_1.
- *restoreActivity*: remove the bypass from a skipped activity.
- *skipConcurrentActivity* is a specialisation of *skipActivity*, which bypasses a path from a concurrent path set. This operator is illustrated in fig.4, being used to credit (skip) the low-fat diet subworkflow.

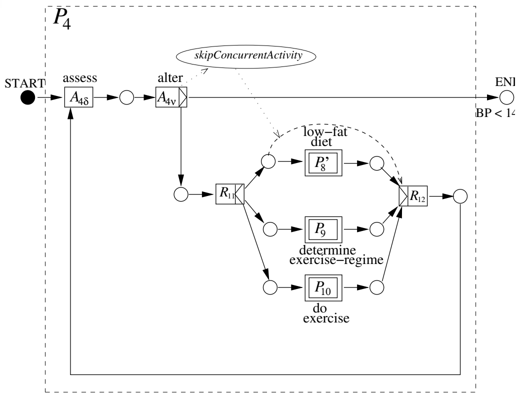

Figure 4: applying a crediting operator to skip lowfat diet

- *restoreConcurrentActivity*: remove the bypass from a skipped activity belonging to a set of concurrent activities.
- *skipChoiceActivity* is a specialisation of *skipActivity*, which bypasses a path from an exclusive-OR (choice) path set. If this operator is temporally

constrained, it may need to be reinstated at a later date/time. E.g.

$$skipChoiceActivity(A_2)$$
$$Unless\ hypertensive(P)\ VALID{-}TIME\ now,$$

The above formula states that activity A_2 should be skipped unless patient P has become hypertensive.

- *restoreChoiceActivity*: remove the bypass from a skipped activity belonging to a set of alternative activities.
- *deleteActivity*: delete a named activity from the workflow schema for this case.
- *deletePostcondition*: delete a data item required as an output of a specific activity. e.g.

  $$deletePostcondition(A, parameter{=}BP),$$

  where A is the activity, and $parameter = BP$ represents the requirement to collect the patient's blood pressure through this activity.
- *addPostcondition*: add a data item required as an output of an activity.
- *addTempPostcondition*: temporarily add a data item required as an output of an activity.
- *deleteTempPostcondition*: temporarily remove a data item required as an output of an activity.
- *alterPostcondition*: alter a postcondition on an activity.
- *disablePrecondition*: disable a precondition on an activity.
- *enablePrecondition*: enable a precondition on an activity.
- *alterPrecondition*: alter a precondition on an activity. This might be used, for example to relax or further constrain a data value, required for a particular activity. e.g.

  $$alterPrecondition(A, parameter{=}BP, condition{=}''BP{>}140/85'')$$

5 IMPLEMENTATION

Activity crediting relies on participants' understanding of the entire care process or workflow schema for each patient. Good process monitoring tools are essential for this understanding, and for each activity, or change in workflow, a snapshot of the case, including rationale for any changes, needs to be available to all relevant participants. An annotated runtime view of the goal hierarchy can be presented to clinicians as a synoptic view of the case, showing which goals have been achieved, supplemented with times and durations.

We have based our implementation methodology on a prototype workflow engine that extends several of The Workflow Management Coalition's Application Programming Interfaces (API) (Fischer, 2001), by providing support for goal views through a Goal Definition Language and Goal to Process Transformer. We provide support for runtime workflow schema alteration through Event/Condition/Action (ECA) rules similar to those developed for the ACTIVETFL framework used in the workflow management system AGENTWORK W*f*MS (Müller, 2002). AGENTWORK has been applied in HEMATOWORK(Universität Leipzig, 2003) for the management of hemato-oncology, which covers the diagnosis, therapy and follow-up of cancer patients suffering diseases of the hematological and lymphatic node system.

6 CONCLUSION

In this paper, we have outlined the requirements, and introduced a methodology for addressing the duplication of services that might occur in business processes in complex domains such as healthcare. We are currently implementing this approach in a prototype W*f*MS being developed specifically to support goal-based flexible workflow schemas.

Several other issues should be considered when implementing activity crediting mechanisms in practice. These include cost implications and resource recovery. When crediting one activity against the other, for partial or complete elimination of one activity from the workflow, it may be important to compare the cost of each activity, and to consider this in determining which activity might be (partially) skipped. The efficiency motivation for activity crediting is premised on the saving of resources. Any recovery of resources no longer required to service activities that have been credited, should itself be handled efficiently to maximize the advantage of such recovery.

ACKNOWLEDGEMENTS

This work is supported by Australian Research Council SPIRT (Strategic Partnership with Industry Research and Training) grant C00107117 in partnership with the South Australian Department of Human Services.

REFERENCES

Browne, E., Schrefl, M., and Warren, J. (2003). A two tier, goal-driven workflow model for the healthcare domain. In *5th International Conference on Enterprise Information Systems*.

Browne, E., Schrefl, M., and Warren, J. (2004). Goal-focused self-modifying workflow in the healthcare domain. In *37th Hawaii International Conference on System Sciences (HICSS37)*. to appear.

Dardenne, A., van Lamsweerde, A., and Fickas, S. (1993). Goal-directed requirements acquisition. *Science of Computer Programming*, 20(1-2):3–50.

Ellis, C., Kling, R., Mylopoulos, J., and Kaplan, S., editors (1995). *Dynamic change within workflow systems*, Milpitas, California. ACM SIGOIS, ACM Press, New York.

Field, M. and Lohr, K., editors (1990). *Clinical Practice Guidelines: Directions for a New Program*. National Academic Press, Washington, DC.

Fischer, L. (2001). Workflow handbook 2001. Technical report, Workflow Management Coalition (WfMC).

Gross, D. and Yu, E. (2001). Evolving system architecture to meet changing business goals: an agent and goal-oriented approach. In *ICSE-2001 Workshop: From Software Requirements to Architectures (STRAW2001)*, pages 13–21, Toronto, Canada.

Jacobs, S. and Holten, R. (1995). Goal-driven business modelling: Supporting decision-making within information systems development. Technical report, RWTH, Aachen.

Manolescu, D.-A. and Johnson, R. E. (1999). Dynamic object model and adaptive workflow. OOPSLA'99 Metadata and Active Object-Model Pattern Mining Workshop.

Miksch, S., Shahar, Y., and Johnson, P. (1997). Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. In *7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97)*, Milton Keynes, UK.

Müller (2002). *Event-Oriented Dynamic Adaptation of Workflows: Model, Architecture, and Implementation*. PhD thesis, Fakultät für Mathematik und Informatik der Universität Leipzig.

Mylopoulos, J., Chung, L., and Yu, E. (1999). From object-oriented to goal-oriented requirements analysis. *Communications of the ACM*, 42(1):31–37.

Reichert, M. and Dadam, P. (1998). ADEPT flex - supporting dynamic changes of workflows without losing control. *Journal of Intelligent Information Systems*, 10(2):93–129.

Sadiq, S. and Mangan, P. (2002). On building workflow models for flexible processes. In *Thirteenth Australasian Database Conference*, volume 5, Melbourne, Australia. Australian Computer Society, Inc.

Snodgrass, R. and Ahn, I. (1985). A taxonomy of time databases. In *Proceedings of the 1985 ACM SIGMOD international conference on Management of data*, pages 236–246.

Universität Leipzig (2003). Hematowork: A workflow system for protocol-directed care in distributed hemato-oncology. http://dbs.uni-leipzig.de/de/Research/hemato.html.