

Semi-Automated Workflow Synthesis

Abilio Fernandes, Karin k. Breitman, Tatiana A.S.C. Vieira, Marco A. Casanova, Antonio L. Furtado
Pontifícia Universidade Católica do Rio de Janeiro
{karin, abilio, tati, casanova, furtado}@inf.puc-rio.br

Abstract

Workflow management systems usually interpret a workflow definition rigidly. However, there are real life situations where users should be allowed to deviate from the prescribed static workflow definition for various reasons, including lack of information, unavailability of the required resources and unanticipated situations. Furthermore, workflow complexity may grow exponentially if all possible combinations of anticipated scenarios must be compiled into the workflow definition. To flexibilize workflow execution and help reduce workflow complexity, this paper proposes a dual strategy that combines a library of predefined typical workflows, of manageable complexity, with a planner mechanism capable of synthesizing more complex workflows, at execution time, tuned to the scenario in question. This dual strategy is motivated by the difficulty of designing emergency plans, modeled as workflows, which account for real life complex crisis or accident scenarios..

1. Introduction

Workflow management systems have been receiving considerable attention lately, motivated by their wide spectrum of applications. Standardization efforts are under way in the context of consortia, such as WfMC, OASIS and W3C. In particular, WfMC was created in 1993 by several companies with the purpose of standardizing workflow concepts and technologies. Among other contributions, these efforts resulted in new workflow definition languages and coordination protocols.

Requirements for workflow management systems comprise a long list. Among the requirements, we may highlight distributed execution, cooperation and coordination, and synchronization, which influence the way user communities work cooperatively to perform a given task.

This paper addresses what we collectively call workflow flexibilization. Briefly, workflow management systems usually interpret a workflow definition rigidly. However, there are real life situations where users should be allowed to deviate from the prescribed static workflow definition for various reasons, including lack of information and unavailability of the required resources. Furthermore, in complex domains, such as crisis response

planning, designing a single workflow that accounts for all possible situations may become an overwhelming task. The final workflow may become too complex to be amenable to formal verification and may therefore not cover all desired scenarios.

To achieve workflow flexibilization, we propose in this paper a mixed strategy, combining a library of predefined workflows, of a limited degree of complexity, with a planner component that, based on the observed complex scenario, searches through the library, retrieves the appropriate workflows and combines them into more complex units that can handle the observed scenario. We call this strategy mixed in the sense that it starts with pre-compiled plans, stored in the form of workflows, to create more complex plans, at run time, with the help of a planner component.

The next two sections expand the discussion about the motivations for our approach and outline the proposed strategy, including a remark on workflow verification.

2. Motivation

Consider the problem of designing emergency plans for an oil terminal located in a harbor facility. Examples of accidental scenarios are: vessels docked at the harbor may catch fire, causing explosions followed by oil spills; vessels may hit underwater oil pipelines causing oil spills of a different nature; inland oil tanks may catch fire and produce harmful pollution reaching nearby communities.

The company that operates the oil terminal must create emergency plans to account for multiple accidental scenarios. The local environmental protection agency, and other government authorities, audits the company from time to time to verify if the company has emergency plans that cover the plausible accidental scenarios, and if it has the necessary equipment or if it has access to the required resources within a reasonable amount of time.

For example, consider the problem of cleaning coastal areas affected by an oil spill. To address this type of accident, the company may specify an emergency plan that defines a set of cleaning procedures that take into account the oil type and the characteristics of the coastal area. Table 1 provides a schematic example of cleaning procedures, where the type of coastal area is Sand Beach and the oil types are named Type I through Type V. The weights in the table cells indicate the environmental impact of each procedure: 0.00 indicates the smallest environmental impact, 0.25 some impact, 0.50 a

significant impact, 0.75 the greatest impact, and 1.00 inapplicable.

Now, suppose that an emergency team is assigned to the accident. This team will be referred to as the user of the emergency plan in what follows. Suppose that the user comes to a point in the overall emergency plan execution where he needs to select a cleaning procedure for a sand beach affected by an oil spill. The user can then look up in Table 1 to select the best procedure to clean the beach. For example, if the oil is of Type II, the best procedures are “VC: Vacuum Cleaning” and “CL: Cold, Low Pressure Cleaning”. Based on the equipment available, he may then select a feasible procedure and proceed to execute it. This sequence of steps has to be repeated for each coastal segment that the oil spill affects. Note that the cleaning procedures selected for the various coastal segments may interfere with each other, if they use the same equipment, or if the personnel available are limited. The user would then have to schedule the cleaning procedures within the bounds of the resources available. The above example covers just a simple scenario, i.e., a single accident – an oil spill. The problem lies in that an accident may trigger a sequence of scenarios, such as an explosion, followed by fire and by an oil spill. Therefore, emergency plans must not be too rigid, or else they become too complex to accommodate all possible complex scenarios, generated by multiple accidents. Furthermore, as already pointed out, the amount of resources available – equipment, personnel, etc. – constrain the number of sub-plans that can be executed in parallel.

Our earlier experience in designing a crisis response information system indeed indicates that emergency plans should be synthesized on demand, based on the current accidental scenario, from a library of reasonably small workflows that address simple scenarios. This approach proved to be a feasible strategy to beat the combinatorial explosion that grows out of complex accidental scenarios. Sophisticated plans that try to account for such complex scenarios become too large to be amenable to verification, a mandatory requirement of the government agencies.

Cleaning Procedures \ Oil Type	Type I	Type II	Type III	Type IV	Type V
ND: Natural Degradation	0.00	1.00	1.00	1.00	1.00
UA: Use of Absorbents	1.00	0.25	0.00	0.00	0.00
VC: Vacuum Cleaning	1.00	0.00	0.00	0.00	0.00
CL: Cold, Low Pressure Cleaning	1.00	0.00	0.00	0.25	0.25
CH: Cold, Low Pressure Cleaning	1.00	0.25	0.25	0.25	0.25
HL: Hot, Low Pressure Cleaning	1.00	1.00	0.50	0.50	0.50
HH: Hot, Low Pressure Cleaning	1.00	1.00	0.50	0.50	0.50
PC: Vapor Cleaning	1.00	1.00	0.75	0.75	0.75

Table 7.1 Environmental impact of cleaning procedures for sand beaches.

3. Semi-automated Workflow Synthesis

To achieve workflow flexibilization, we propose in this paper a mixed strategy, combining:

- a library of pre-defined workflows, which address specific scenarios;
- a planner component that synthesizes more complex workflows, based on the observed complex scenario.

Briefly, we start with an appropriate subset of first-order languages used to specify what we call *simple* and *complex scenarios*. For example, given an alphabet consisting of a finite set of variables, $\{x_1, \dots, x_m\}$, and a finite set of constants, $\{d_1, \dots, d_n\}$, we define a *simple scenario* as a conjunction of literals of the form $x_j = d_k$ or $\neg (x_j = d_k)$, and a *complex scenario* as a disjunction of simple scenarios.

A *workflow* is specified in a subset of OWL-S process ontology and has a *pre-condition* and *effects* both defined as complex scenarios. The subset of OWL-S is chosen in a way that permits formal verification of workflows. As in OWL-S, an *abstract workflow* is a workflow that cannot be directly executed because it does not have any associated implementation. Hence, an abstract workflow, by itself, cannot generate any workflow instances. By contrast, a *concrete workflow* has an associated implementation and, consequently, may generate workflow instances. Note that abstract workflows offer additional flexibility in that they allow the designer to create a workflow whose final specification is left for the workflow execution module to complete.

Given a pair of complex scenarios, (I, O) , we say that a workflow W *satisfies* (I, O) iff the pre-condition of W is implied by I and the effects of W imply O .

A *workflow library* is a set of workflows, abstract or concrete, possibly augmented with a *ranking function* that assigns weights to the workflows. A ranking function is necessary when the library contains multiple workflows with the same pre-conditions, or at least non-disjoint pre-conditions, as illustrated in Table 1. The workflow library also has a *matching function* that helps retrieve (a ranked list of) workflows whose pre-conditions are satisfied by a given scenario.

A *planner* for a workflow library WL is an algorithm that, given a pair of complex scenarios, (I, O) , creates a workflow (in the subset of OWL-S chosen) that satisfies (I, O) . The planner uses the workflows stored in WL as building blocks.

The challenge, addressed in the full paper, is to limit the subset of FOL used to specify the scenarios, and the subset of OWL-S, selected to define workflows, so that the following question is decidable (in reasonable time): “Let WL be a workflow library. Given a pair of complex scenarios, (I, O) , will the planner for WL create a workflow that satisfies (I, O) ”? This is essentially the

question government agencies ask when auditing companies, as explained in the previous section. Finally, the subsets of FOL and OWL-S must be expressive enough to cover (most of) real life situations.

Acknowledgements

This work is partially supported by CNPq under grants 550250/05-0, 551241/05-5 and 552068/02-0.