

USING DISTRIBUTED CSPs TO MODEL BUSINESS PROCESSES AGREEMENT IN SOFTWARE MULTIPROCESS

Luisa Parody, María Teresa Gómez-López, Rafael M. Gasca and Diana Borrego
Department of Computer Languages and Systems, University of Seville, Seville, Spain

Keywords: Business process management, Distributed constraint satisfaction problem, Web services.

Abstract: A business process consists of a set of activities which are performed in a coordination way to obtain an objective. Sometimes the definition of this objective using only a classic business processes management is not possible. When the choreography of the processes cannot be defined with a combination of tasks using sequences, conditions, 'xor', 'or' and 'split' control flow patterns, another representation and solution are necessary to be used. This problem makes difficult the decision making in software management projects. In this paper a way to describe a process agreement is described where the execution and the number of tasks execution order of the Web Services cannot be defined. As a case study, the resource distribution in a multiproject development environment is used. In this case, the processes have to achieve an agreement in function of the business rules that relate the processes. In order to achieve this objective, the Distributed Constraint Satisfaction Problems are used to model and solve this type of problems.

1 INTRODUCTION

Business Process Management (BPM) has received considerable attention since it permits to combine different tasks to obtain a common objective. A business process consists of a set of activities that are performed in coordination in an organizational and technical environment (Weske, 2007). BPM includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes. The bases of BPM are the explicit representation of business processes with their activities and the execution constraints between them. However, the aspect analyzed in this paper is when the execution order and the constraint relation between the processes cannot be described with a traditional business process model in an statical way. The execution order of the activities will be defined in function of the data flow for the different instances. For example if the outputs of a business process A affect to another process B, and the output of process B affects to business A and there is no order between them. In this case, the processes have to achieve an agreement in function of business rules that relate the processes. They have to share a resource to obtain the maximum benefit.

The combination of processes may cause the use of services with private and public information, in

such a way that the decisions cannot be made in an external or centralized way to the processes. That is, the information is distributed in different nodes although all of them have to work together. This type of Business Process behavior has some aspects in common with DisCSPs hence we think that some of the advantages of DisCSPs can be adapted to solve the Business Process Agreement.

A company specialized in Custom Software Development Products tends to work in a multiproject environment. The complexity inherent in a software development management increases in the presence of several concurrent projects which are performed by the same organization that must allocate resources between them. (Navascués Fernández-Victorio and Ramos Román, 2008) presents a basic scheme of business process to this type of companies. The organization plans the production of a project by means of the assignment of resources based on its capacity, the agreements acquired with clients and the global workload. In other words, based on the business agreements for each project. In addition, the company has an internal control system to manage the execution and change the assignment to deal with internal or external unexpected events in the projects. It is difficult to determine when the process will end since all its features cannot be known "a priori".

Moreover, the company has to establish the dead-

lines of the projects with each client since the planning is very important to the success or failure of the project. The resources management is crucial since the concurrent projects compete for the resources existing within the company. To reach a solution, they have to achieve an agreement ensuring their interests and giving in some cases. Each project has its own contract with client established by the business rules, where time delay and final cost of the project are related.

The structure of the paper is as follows: Section 2 includes some relevant proposals. Section 3 presents some basic concepts related to DisCSP and its Asynchronous Backtracking Algorithm. Section 4 describes how the algorithms for DisCSPs can be adapted to our problem. Section 5 shows an example of combination of Web Services and describes how our proposal is applied to the example. Section 6 depicts the experimental results. Finally conclusions and future work are presented.

2 RELATED WORKS

The graphical Standard for modeling business processes is Business Process Modeling Notation (BPMN) proposed by OMG. However, BPMN standard is not powerful enough since, among other issues, there is an important need of many adaptive processes for concurrent activities to respect coordination constraints. These require that concurrent activities coordinate their behaviors in response to exogenous events. In order to represent the choreography of the processes in a business process, several graphical languages have been developed. The most important is the standard BPMN (Weske, 2007), that can be used for large number of problems (Wolter and Schaad, 2007).

3 DISTRIBUTED CONSTRAINT SATISFACTION PROBLEM

In (Yokoo and Hirayama, 2000), Makoto et al. presented a Distributed Constraint Satisfaction Problem (DisCSP) as a general formalism for dealing with problems in multi-agent systems. A DisCSP is a CSP where the set of variables and constraints of the problem are distributed between a set of agents. These agents are in charge to solve their own sub-problem and must coordinate themselves with the rest of agents to reach a solution to the global problem (Abril López et al., 2007).

In order to improve the time to find the best solution, the paper of Redouane Ezzahir et al. (Ezzahir et al., 2008) proposes algorithms for solving Distributed Constraint Optimization Problems (DCOP).

3.1 Backtracking Algorithms to Solve DisCSPs

There are different proposals to model and solve DisCSP:

Centralized Backtracking selects a leader agent among all agents who gather all the information about variables, their domains, and their constraints (Yokoo and Hirayama, 2000). If the knowledge about the problem can be gathered into a single agent, this agent can solve the problem alone by means of the use of normal centralized constraint satisfaction algorithms.

Asynchronous algorithms are characterized by the fact that all agents are active in parallel and only coordinate as needed to ensure consistency if their variables are involved in the constraints.

In Asynchronous Backtracking (ABT), the priority order of variables/agents is determined, and each agent communicates its tentative value assignment to its neighboring agent via 'ok?' messages. Each agent maintains the current value assignment of other agents from its viewpoint. An agent changes its assignment if its current value assignment is not consistent with the assignment of higher priority agents. If there exists no value that is consistent with the higher priority agents, the agent generates a new constraint (called a *nogood*), and communicates the *nogood* to a higher priority agent, thus the higher priority agent changes its value making backtracking.

In (Yokoo and Hirayama, 2000) the soundness, completeness and termination of ABT algorithm is proved.

3.1.1 Heuristics for Variable Ordering

There are different forms to order the variables to establish the priority: (a) The highest priority is assigned to the variable that participates in more constraints, called variable degree. There are different possibilities to establish the next highest priority variable: (a.1) The next variable that participates in more constraints, without considering the agent who it belongs to. (a.2) The next variable that participates in more constraints and belongs to the same agent that the first one. (b) The highest priority is assigned to the variables that belong to the fastest agent. (c) The priority is assigned according to the problem specification.

4 USING DISCSP TO MODEL BUSINESS PROCESS AGREEMENTS

Business Process Execution Language for Web Services (WS-BPEL) is emerging as the prominent language for Modeling Executable Business Processes. This is due to WS-BPEL provides an appropriate set of constructs to design a process in an intuitive way. One of its features is the ability to synchronize concurrent flows within a process by specifying control links between activities. There could be additional types of constraints between concurrent activities. The constraint that involves the variables of the different agents, implying their equality, requires actions in response to events in order to be coordinated. Such types of constraints between concurrent activities are called *coordination constraints* (Wu and Doshi, 2008). These constraints must adapt to events to preserve their optimality. Unfortunately, *coordination constraints* are not natively supported by WS-BPEL, its support for designing flexible processes is limited.

On the other hand, the aim of these types of problems is to obtain a common objective, which depends on the business agreements of each agent, the client and on the contract established through business rules.

The activities that need this type of coordination to obtain the overall goal have some aspects in common with the features of DisCSPs discussed in Section

3. In this paper, our proposal adapts DisCSPs to the agreement problems between tasks to obtain the objective. As the algorithm based on the algorithms presented in Section 3.1, we have designed a new algorithm to make Business Processes. It can be classified as a mixture of Centralized and Asynchronous Backtracking. It is centralized since the Business Process has the knowledge about the values of the variables, and it is asynchronous since all the agents are active in parallel and communicate their consistencies or inconsistencies asynchronously. The key is that the algorithm invokes to an activity that calculates the objective. Then, we translate this algorithm into a Business Process (Figure 1).

The most important activity in the BP (Figure 1) is *Calculate OBJECTIVE* activity, where each problem models its way to obtain the overall goal.

5 MOTIVATING EXAMPLE

As an example to illustrate our proposal, a Web Service to perform the assignment of people to tasks presented in the introduction (Section 1) is used. During the development of a software product, different tasks or activities take place, giving rise to the software life-cycle.

Typically, a lifecycle model addresses the phases of a software project: requirements, design, implementation, integration, testing, operations and maintenance. In our example, the company has to deal with the performing of the implementation phase of some different projects concurrently. The staff is made up of different number of workers on each different professional category, who must be assigned properly to perform the tasks of these projects. The profits obtained by the company depends on the delay of project. The fulfillment of the delivery time agreed with the customer provides the maximum profits for each project. An optimal staff distribution will do the sum of profits as maximum as possible. The calculation of the profits of each project is internally made within each project separately, so that it is necessary to get an agreement to achieve the overall goal. For example, we have three projects. If two workers are assigned to project 1, it implies that the profit will be 1 million euros and for project 2, the profit will be 0.4 million. But, if these two workers are assigned for project 2, the profits will be 0.8 and 0.7 million for project 1 and 2 respectively, so this is the best assignment to obtain the maximum global profit.

The profits that the company obtains by each project depend on the business agreement between each project manager, the project's client, and by the

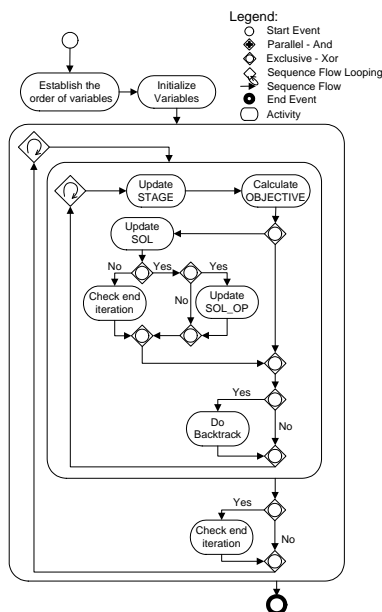


Figure 1: Backtrack Business Process.

contract established through business rules. The calculation of the profits of each project is performed in a different Web Service. The combination of those calculations will provide the overall profit to maximize.

5.1 Backtracking Algorithm applied to the Motivating Example

In the particular case of the Software Management Project, there are three different projects that work concurrently and need some kind of staff organization. Although these projects belong to the same company, they do not have to reach the same agreements with clients or the internal contracts. For security and privacy reasons, they share only the necessary information to achieve the overall goal. Each project will correspond to an agent. These agents have to agree to maximize the profit of the company, trying to finish their projects on time with the available staff.

In this section, the proposal presented in Section 4 is applied to the problem of Software Management Projects.

Considering the cases studied in Section 3.1.1 and the presented problem, all the variables are involved in the same number of constraints and there are no significant differences in the speed of response time of the different agents. In this case, to establish the priority order it is necessary to take into account the overall goal of the problem: to maximize the overall profit. The variables which belong to the project which gives the greatest profit will have the highest priority. With this criterion, the algorithm may find a good solution quickly and make more efficient bounds.

The most important activity in the BP (Figure 1) was the *Calculate OBJECTIVE* activity, in the motivating problem this activity is *Calculate PROFIT*. In this activity there are such parallel flows belonging to each agent. Each flow has two activities: (a) *Solve variables of the agents*: it uses the constraints that the agent has associated to assign consistent values to variables. (b) *Establish the profit*: it calls the Web Service which calculates the profit with the values assigned to variables in the previous activity.

6 EXPERIMENTAL RESULTS

Table 1 shows the staff distribution given the following input: 5, 10 and 6 members of staff with categories 1, 2 and 3 respectively. The profit obtained with this distribution is 840500 euros.

Table 1: Results.

Variables	P1	P2	P3
num. of T1	4	0	1
num. of T2	8	1	1
num. of T3	1	5	-

7 CONCLUSIONS AND FUTURE WORK

In this work, a Business Process that adapt the Backtracking Algorithm for DisCSP to Business Process is presented. This proposal arises from the need to coordinate different activities that belong to a Business Process and work concurrently sharing resources. Thanks to this adaptation, the activities achieve the coordination to obtain an overall goal, which, in the example presented in this paper, is the software multi-project profit.

One of the main problems in the search of solutions is the size of the search space. A bound could be established to avoid calling agents and calculate the profit if we know that we are not going to get a better result. We could obtain the bound from: (i) the search space that is already done and (ii) the "post-mortem" memory (Navascués Fernández-Victorio and Ramos Román, 2008) of previous projects.

ACKNOWLEDGEMENTS

This work has been partially funded by the Junta de Andalucía by means of la Consejería de Innovación, Ciencia y Empresa (P08-TIC-04095) and by the Ministry of Science and Technology of Spain (TIN2009- 13714) and the European Regional Development Fund (ERDF/FEDER).

REFERENCES

Abril López, M., Barber Sanchs, F., and Salido Gregorio, M. A. (2007). *Particionamiento y Resolución Distribuida Multivariable de Problemas de Satisfacción de Restricciones*. PhD thesis, Universidad Politécnica de Valencia, Valencia.

Ezzahir, R., Bessiere, C., Benelallam, I., Bouyakhf, H., and Belaïssaoui, M. (2008). Dynamic backtracking for distributed constraint optimization. In *ECAI*, pages 901–902.

Navascués Fernández-Victorio, J. and Ramos Román, I. (2008). *Un modelo para la simulación híbrida de la producción de software a medida en un entorno multiproyecto*. PhD thesis, Universidad de Sevilla.

- Weske, M. (2007). *Business Process Management: Concepts, Languages, Architectures*. Springer.
- Wolter, C. and Schaad, A. (2007). Modeling of task-based authorization constraints in bpmn. In *BPM*, pages 64–79.
- Wu, Y. and Doshi, P. (2008). Making bpel flexible - adapting in the context of coordination constraints using ws-bpel. pages 423–430.
- Yokoo, M. and Hirayama, K. (2000). Algorithms for distributed constraint satisfaction: A review. In *AAAI*, volume 3, pages 198–212.

