Determination of an Optimal Test Points Allocation for Business Process Analysis

Diana Borrego*, María Teresa Gómez-López*, Rafael M. Gasca* and Rafael Ceballos*
*Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla, Sevilla, Spain
Email: {dianabn, maytegomez, gasca, ceball}@us.es

***Abstract*—The management and automation of business processes have become an essential task within IT organizations. Diagnosis enables fault isolation in a business process. The diagnosis process uses a set of test points (observations) and a model in order to explain a wrong behavior. In this work, a series of algorithms to allocate test points are presented. The key idea is to improve the diagnosability, improving the computational complexity for isolating faults in a system. The methodology is based on constraint programming.**

***Index Terms*—business processes, constraint satisfaction problems, fault diagnosis, fault isolation, test points.**

I. INTRODUCTION

A business process (BP) is composed of a set of activities which are logically related to achieve a defined goal. BPs can be composed of different subprocesses and a large number of activities that interact by means of a choreography with the same process or another.

Business process management includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of BPs [1].

Fault diagnosis permits to determine why a BP correctly designed does not work as it is expected, detecting and identifying the reason of an unexpected behavior. It is based on observations, which come from the public information existing in the BP to diagnose, that is measured either by the direct observation of the user or by means of test points allocated in certain places of the BP. The appropriate allocation of test points makes possible the isolation of faults in activities of a BP when the diagnosis is performed. The aim of this work is to apply techniques of allocation of test points in BPs, in order to improve the computational complexity of isolating faults in the diagnosis process and make the BP diagnosable.

In order to allocate test points, a BP can be modelled as a Constraint Satisfaction Problem (CSP) ([2]), where the topology of the BP is transformed into a graph, in such a way that the nodes and edges can be modelled using constraints.

The paper is structured as follows: Section 2 explains how it is possible to solve the problem of the allocation of test points in BPs, including three different objectives to achieve. Section 3 shows experimental results. And finally, conclusions and future work are presented.

II. ALLOCATION OF TEST POINTS IN A BP

The aim of this paper is to apply techniques for the allocation of test points. Those test points are allocated to read private information that is exchanged between activities of a BP. This paper is centered in the BPs that fail for any instance or its behavior does not correspond with the expected.

The test points make possible the separation of the activities into different clusters, reducing the computational complexity in the diagnosis process and making easier the detection of the incorrect activity. This is possible since the diagnosis of the whole BP can be performed based on the diagnosis of each cluster separately.

When it comes to allocate test points in a BP, the algorithm proposed can be configured to achieve different objectives. In this paper, the presented algorithm can be configured to solve the problem with three different objectives, depending on the user necessity or the problem requirements.

The implemented algorithms and these different objectives to achieve are detailed in the following. Since the BPs are going to be modelled as CSPs, first it is going to be introduced how the graph is transformed.

A. Modelling and solving the problem as a CSP

A CSP is a problem to find a consistent assignment of values to variables. In order to allocate test points, the problem will be modelled as a CSP.

Initially, the BP is considered as a directed graph, where the activities are the nodes and the connections between them are the edges. Those nodes and edges are translated into variables in the CSP and the topology of the graph gives rise to constraints between those variables, establishing that if there is no a test point in a link of two activities, both activities have to be in the same cluster. It is not possible to assert the opposite statement, since the existing of a test point in a connection between two activities cannot imply that they are in different clusters since it is possible that they are connected through another path in the graph.

Those added constraints are common to the three objectives that the solution proposed in this paper achieves. But each objective needs different new constraints and goals to be completely modelled.

B. Objective 1: to maximize the number of clusters with a fixed number of test points to allocate

In order to achieve this objective, new information must be added to the CSP: the number of test points is limited to a fixed value and the specific goal is included.

978-1-4244-6039-7/10/$26.00

The computational complexity of this CSP is exponential for the number of connections: with a large number of activities, the time needed to solve the CSP makes this solution inappropriate. To sort out this problem, the greedy algorithm presented in [3] is used. That algorithm assigns a weight to each edge, and applies the Floyd's algorithm to find the minimal path between each pair of nodes. Those paths decide through a voting mechanism which are the most important connections to allocate test points on them. The obtained result is a set with the different connections of the BP and the votes received for each one of them. This set is used to select the collection of connections that will be better to allocate test points on them. The number of selected connections depends on the user or the problem specification. Those connections will be the only ones taken into account in the solution of the CSP previously explained, improving the computational complexity. If only n connections are chosen, the number of possible solutions of the CSP is lower than 2^n. On the other hand, the optimal solution is not guaranteed.

C. Objective 2: to allocate the minimum number of test points in order to obtain a fixed number of clusters

In order to model this solution, it is necessary to include some constraints on the CSP. They establish the number of clusters in a fixed value determined by the user or the specification of the problem. The configured goal is to minimize the number of test points. This CSP does not present computational problems, so that it is not necessary to include any previous method to improve the execution time.

D. Objective 3: to minimize the number of test points in order to obtain clusters with a maximum number of activities

In order to model this objective, it is necessary to add more constraints to the initial CSP: to limit the number of activities that belong to each cluster, or to keep the CSP solver from finding out equivalent solutions. In this case the goal to achieve is an objective function to establish the minimization of the number of test points to allocate.

When the CSP solver is executed to find the optimal solution, the computational complexity of the problem is exponential, so that it is necessary to add some kind of bound to reduce the search space of the variables in the CSP.

In order to get a bound, a new greedy method is used. That method allocates test points in the BP in a linear time. The solution provided by this new greedy algorithm may not be the optimal solution, but it is close to the optimal, and provides a very useful bound for the number of test points to allocate that reduces drastically the domain of the variables.

The greedy algorithm is based on the topology of the BPs, taking advantage of the different control flow patterns that are used to model a BP. Since frequently in BPs there are topologic structures where a set of branches that form a split are synchronized by means of a join, it is possible to analyze the processes in a deep way. Those splits and joins will enable to divide the BP in different levels. This is, when a single thread of execution splits into two or more branches, and those branches converge in a join, the activities in those branches are in an inferior level than the activities in the main thread.

Based on this idea of levels, the greedy algorithm is made up of several steps: transformation of the BP into a graph, labelling of the nodes and allocation of the test points.

III. Experimental results

In this section, the computational complexity of the exhaustive and greedy method in Objective 3 are compared. We present the execution time of the explained algorithms applied to some BPs with different number of activities (from 5 to 50 activities per BP). Those BPs are benchmarks that have been generated to check the three different objectives to achieve.

Fig. 1 shows the difference between the execution time spent by the exhaustive and the greedy method explained to solve Objective 3. It is possible to see the difference between the exponential execution time for the exhaustive method and the linear complexity when the greedy algorithm is used to establish a bound in the number of test points.

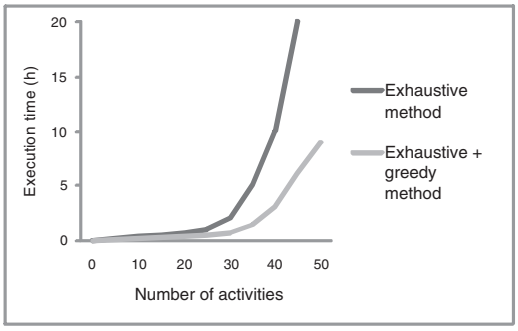

Fig. 1. Execution time for Objective 3.

IV. Conclusions and future work

This work presents a series of algorithm to allocate test points in BPs. The aim is to apply techniques of allocation of test points in order to improve the computational complexity of isolating faults in the diagnosis process, and make the BPs diagnosable. Three different objectives have been achieved, depending on the user necessity or the problem specification.

As future work, it is interesting to perform the diagnosis of the BPs once the test points have been allocated.

Acknowledgements

This work has been partially funded by Junta de Andalucía by means la Consejería de Innovacin, 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

[1] M. Weske, "Business process management. concepts, languages, architectures," *Springer*, 2007.

[2] F. Rossi, P. van Beek, and T. Walsh, Eds., *Handbook of Constraint Programming*. Elsevier, 2006.

[3] R. Ceballos, V. Cejudo, R. M. Gasca, and C. D. Valle, "A topological-based method for allocating sensors by using csp techniques," in *CAEPIA*, ser. Lecture Notes in Computer Science, R. Marín, E. Onaindia, A. Bugarín, and J. Santos, Eds., vol. 4177. Springer, 2005, pp. 62–68.