Analysis of Resource Allocation of BPMN Processes

Francisco Durán[1(✉)], Camilo Rocha[2], and Gwen Salaün[3]

[1] University of Málaga, Málaga, Spain
duran@lcc.uma.es
[2] Pontificia Universidad Javeriana, Cali, Colombia
[3] University of Grenoble Alpes, LIG, CNRS, Grenoble, France

Abstract. The approach for the modelling and analysis of resource allocation for business processes presented in this paper enables the automatic computation of measures for identifying the allocation of resources in business processes. The proposed analysis, especially suited to support decision-making strategies, is illustrated with a case study of a parcel ordering and delivery by drones that is developed throughout the paper. BPMN models are represented in Maude.

1 Introduction

This work presents first steps towards the development of a formal and automatic approach to resource allocation analysis for business process models. The approach comprises a formal yet executable specification in rewriting logic [3], a logic of concurrent computation, of a significant and expressive subset of the *Business Process Model and Notation* (BPMN) extended with time features and resources. By being executable in the Maude system [1], the specification supports the concurrent simulation of a process with different types of resources and with multiple replicas for any given workload. The analysis techniques for resource allocation use Maude's rewriting tools for evaluating expected values in the executable model — such as charge, occupancy, and usage percentage — by mechanically generating automatic simulations. The output of the automatic analysis can then be used to quantitatively assess the efficiency of the business process model, and thus guide a re-design or re-allocation of resources.

The overall idea is that multiple concurrent executions of a process compete for shared resources. Models are analyzed by observing how the resources' usage evolve over time when varying the workload and the number of available resources. This is done without an implementation of the system running on real resources: the input to the automatic analysis task is a BPMN model of the process workflow, enriched with a description of its timing behavior and resource

F. Durán was partly funded by project *PGC2018-094905-B-I00* (Spanish MINECO/FEDER), and by U. Málaga, Andalucía Tech. C. Rocha was partly supported by Colciencias-EcosNord project "FACTS: Foundational Approach to Cognition for Today's Systems" (63561).

S. Yangui et al. (Eds.): ICSOC 2019, LNCS 11895, pp. 452–457, 2019.
https://doi.org/10.1007/978-3-030-33702-5_35

availability. The BPMN models are described by means of activity and collaboration diagrams, four types of gateways (namely, inclusive, exclusive, parallel, and event-based), loops, unbalanced processes, event handling, and message-passing. The timing aspects of the models are specified by associating durations to each flow and task in the workflow, which can be sampled from a probability distribution function. Resource usage is specified by providing the amount of resources available and the resources required for the execution of tasks.

The usefulness of the approach is illustrated with an experiment, which supports the claim that such an analysis can help in detecting resource usage problems, thus ultimately leading to the improvement of the business process by optimizing its resource allocation. In particular, the experiment presented in this paper identifies low-level occupancy of resources and undesirable patterns of resource usage. They encompass sequential dependencies and bottlenecks provoked by some highly used resources that may induce performance fall-downs.

2 Business Process Model and Notation

Figure 1 presents a process describing a parcel ordering and delivery by drones. This BPMN process is presented as a collaboration diagram consisting of two pools, one for the client and another one for the order management and the delivery process; they are represented as lanes. This process includes different kinds of gateways, probabilities for choice gateways, stochastic functions for time associated to tasks, a loop, and unbalanced structures.

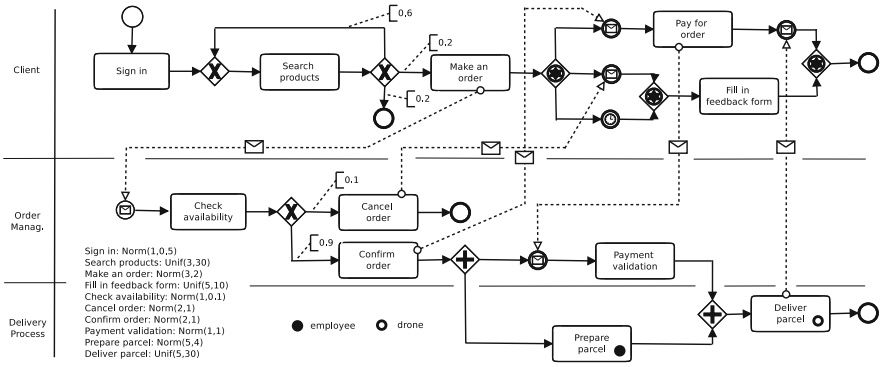

Fig. 1. Running example: parcel delivery by drones.

As usual, start and end events are used, respectively, to initialize and terminate processes. A task represents an atomic activity that has exactly one incoming and one outgoing flow. A task may have a duration (expressed as a stochastic expression) and may produce an event message. A sequence flow describes two nodes executed one after the other, i.e., by imposing an execution order with possible delays.

The timing information associated to tasks and flows (durations or delays) is described either as a literal value or sampled from a probability distribution function. Gateways (*exclusive*, *inclusive*, *parallel*, and *event-based*) are used to control the divergence and convergence of the execution flow. Gateways with one incoming (resp., outgoing) branch and multiple outgoing (resp., incoming) branches are called *splits* (resp., *merges*). Event-based split gateways may have a default branch fired by a timeout. Workflows with looping behavior are supported, as well as unbalanced workflows.

Data-based conditions for split gateways are modeled using probabilities associated to outgoing flows of exclusive and inclusive split gateways. For instance, notice the exclusive split after the Search products task in the Client lane of the running example, which has outgoing branches with probabilities 0.6, 0.2, and 0.2, specifying the likelihood of following each corresponding path.

Instead of implicitly associating resources to lanes, resources are explicitly defined at the task level, which is more general. A task that requires resources can include, as part of its specification, the number of required instances (or replicas) of a resource. The process in Fig. 1 relies on employees for parcel packing and drones for parcel delivery. Notice the colored circles at the bottom-right corner of the Prepare parcel and Deliver parcel tasks, indicating that one instance of the employee resource and another one of the drone resource are required, respectively, for the tasks completion. Several tasks could compete for the same resources (not the case in this example). Furthermore, since multiple instances of a process may be executed concurrently, all instances also access and compete for the shared resources. At the bottom-right corner of Fig. 1, a total of two employees and three drones are specified as the available resources.

3 Resource Allocation Analysis

This section illustrates how resource allocation analysis can be performed with the proposed approach using the running example. Given a process description, a specification of resources, and a workload, the experiments illustrate how information on execution times and resource usage is collected. This information is used to find the optimal allocation of resources that minimizes costs and execution times relative to an optimization goal. The interested reader is referred to [2] for further details on the experiments.

The BPMN subset encoded in Maude is quite expressive and several kinds of properties can be computed, including timing and resource-based ones. These properties are meaningful when executing multiple instances of a process that compete for the shared resources. As for *timing properties*, the approach presented in this paper allows the computation of average execution times (AET) of a process, its variance (Var), and the average synchronization time (AST) for merge gateways, representing the time elapse from the arrival of the first token through one of its incoming flows to its activation. Synchronization times make sense only for parallel and BPMN 1.0 inclusive gateways, since there is no waiting/synchronization time for the other gateways.

The following *resource-based properties* are computed:

- The global time usage of all replicas of each resource R (GTU_R).
- The average GTU of resource R (GTU^1_R).
- The average usage percentage for a resource R (UP^1_R).

To quantify these properties, Maude rewriting capabilities are used in order to simulate and extract analysis results on a given BPMN process.

Table 1. Experimental results for the running example (2 employees, 3 drones)

Num. inst.	AET	Var	AST_{g8}	AST_{ee}	Total time	Resources						Anal. time
						GTU_e	GTU^1_e	UP^1_e	GTU_d	GTU^1_d	UP^1_d	
100	106	72	58	58	326	271	135	41	853	284	87	5 *s*
200	185	134	71	139	670	514	257	38	1892	630	94	26 *s*
400	284	173	98	237	1132	994	497	43	3270	1089	96	189 *s*
800	506	294	145	459	2217	1867	933.6	42	6525	2171	98	1233 *s*
1600	891	473	240	844	4187	3714	1857	44	12428	4142	98	7909 *s*

Table 1 summarizes experimental results on execution times and resource usage on the parcel order and delivery example (Fig. 1). They were carried out on an iMac with 3,2 GHz Intel Core i5 and 8 GB. All simulations were performed assuming a given workload with a number of instances (1st column) and an exponentially distributed interarrival time ($\lambda = 4$). Columns 2–6 contain, resp., the average execution time (AET), its variance (Var), the average synchronization time for the parallel merge at the end of the delivery process lane (AST_{g8}), the average synchronization time for the end events (AST_{ee}), and the total time to complete the execution of all instances. The next six columns show results on resource usage for employees and drones. The final column gives the overall time needed to complete the analysis. All times are logical units, except the ones in the last column that are given in seconds. Other information, such as the duration of each task and the synchronization time of merge gateways, is also collected.

These experiments consist of 100, 200, 400, 800, and 1600 instances for 2 employees and 3 drones. Note that the average execution and synchronization times, as their variance, clearly increase with the number of instances. This is because the more tokens compete for resources, the more time it takes to execute the process and for the tokens to reach the synchronization points. Note the relationship between AET and AST_{ee} times, showing an unbalance between the two lanes: the client lane terminates earlier than the other lane, which exhibits a bottleneck because of the demand on the resources.

The GTU increases with the number of executed instances. These times are particularly interesting because they can be materialized as costs (e.g., cost of a resource, salary of an employee/all employees). In relation with usage percentage (UP), the results indicate that the employees are "underused" since they work

around 40% of the time, in contrast to the drones that are constantly busy and used about 90% of the time for delivering parcels. This may suggest an inappropriate allocation of resources. It is worth observing that, although the number of instances clearly affects all computed times, the results for resource usage (UP) are quite stable and a small number of instances is enough for obtaining a good approximation of these percentages.

Resource allocation impacts execution times (AET) and resource usage (UP) of a process. Figure 2 focuses on average execution time and depicts the results when the number of employees and drones vary for a fixed number of executions (400). The objective here is to reduce the average execution time for completing the process: the quicker the parcel is delivered, the more satisfied the client is. It can be observed that, independently of the number of employees, execution times are not satisfactory with 1 or 2 drones (between 400 and 800 time units). The time becomes reasonable for more than 3 drones (less than 300 time units) and tends to stabilize. It is also worth noting that, given its low usage rate, the number of employees does not impact significantly the execution time. For more than six drones, only going from one to two employees makes a significant impact in the AET values.

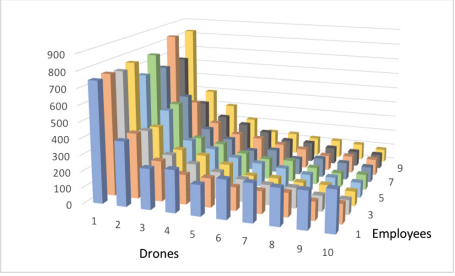

Fig. 2. Average execution time (400 instances)

Figure 3 gives a different point of view of resource usage by concentrating on each resource replica. Figure 3 (left) shows that employees are close to 100% usage only if there is 1 or 2 instances of that resource and at least 4 or 8 drones, respectively. If the number of employees increases, the usage percentage quickly drops, reaching a low level (e.g., 14% for 4 employees and 2 drones). This percentage slightly increases with the number of drone replicas (e.g., 34% for 4 employees and 5 drones), but remains low (around 30%). Figure 3 (right) shows that the drone usage is always quite high whatever the number of employees is. With only 1 or 2 drones, the usage percentage is almost at 100% and slightly decreases with 4 drones. When there are 6 drones and 1 employee, the percentage is still about 60%. Another interesting fact is that the number of employees barely impacts the drone usage percentage. For example, with 4 drones, the usage percentage is around 90% for any number of employees.

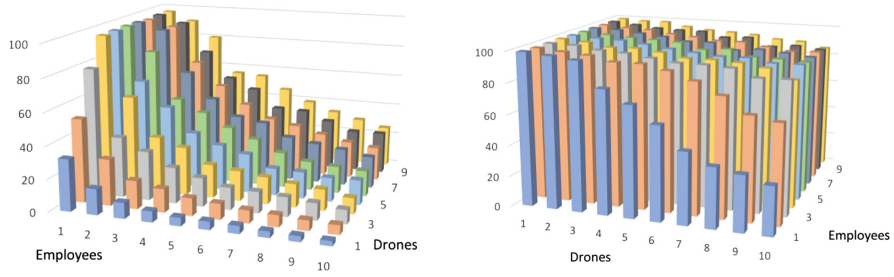

Fig. 3. Average percentage usage per employee (left) and drone (right) (400 instances)

References

1. Clavel, M., et al.: All About Maude - A High-Performance Logical Framework. LNCS, vol. 4350. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71999-1
2. Durán, F., Rocha, C., Salaün, G.: A Note on Resource Allocation Analysis of BPMN Processes, July 2018. http://maude.lcc.uma.es/BPMN-R
3. Meseguer, J.: Conditional rewriting logic as a unified model of concurrency. Theor. Comput. Sci. **96**(1), 73–155 (1992)