BTRANSFORMER
A Tool for BPMN to CSP+T Transformation

Aleksander González[1], Luis E. Mendoza Morales[1], Manuel I. Capel [2], María A. Pérez[1]
Edumilis Méndez[1] and Kenyer Domínguez[1]
[1]*Processes and Systems Department, Simón Bolívar University, PO Box 89000, Caracas 1080-A, Venezuela*

[2]*Software Engineering Department, University of Granada, Aynadamar Campus, 18071 Granada, Spain*

Keywords: Business Process Modelling Language, Process Calculus, Model Transformation Languages, Transformation Tool, Formal Software Specification.

Abstract: In any organisation, properties such as scope, structure, deployment, capability, structural consistency and concurrency, supporting the critical factors for success in Business Process (BP) modelling, need to be verified. And thus, relevant parts of a BP must be formally specified in an appropriate way. Process Calculi (PC) such as CSP, ACP, CCS, which constitute a mathematical basis for programming reactive, communication-bounded systems, can be used to model critical systems and to verify their correctness properties. PC-based notations can be used to specify business processes (BPs) and reason about their properties. Without a demanding training, to make use effectively of these languages is beyond the ability of many business modellers. In order to cope with this drawback, we propose a set of rules to automatically transform a semi-formal model expressed in terms of Business Process Modelling Notation (BPMN) into a Communicating Sequential Processes + Time (CSP+T) formal system specification. In this paper, we present BTRANSFORMER tool that permits to automatically generate such a formal specification and has been programmed with the ATLAS Transformation Language (ATL). As result, we obtain a plug-in for Eclipse platform, which is capable of transforming BPMN models designed with Intalio into a text file with the equivalent CSP+T formal specification of the business model.

1 INTRODUCTION

Business Processes (BP) models are of use to represent a set of properties of interest to organizations, such as scope, capability planning, process general structure, data dependency, safety and concurrency (Weske, 2007; Aalst y Hee, 2004). These properties can be derived from the analysis of a BP model built using a semi-formal language, such as BPMN (OMG, 2009) or UML. In order to verify that a model of a BP satisfies the aforementioned properties, it is possible to resort to formal specification languages, capable to represent the abstract execution of the model, and to verification tools based on these notations. Communicating Sequential Processes (CSP) (Hoare, 1985) is a formal language based on a Process Calculus (PC) of processes that communicate. Verification tools, such as FDR (Formal Systems (Europe) Ltd., 2005) based on CSP, allows modelling concurrent systems in a structured way, as well as to represent some model's behaviours and abstract non-relevant data to the business interaction level.

The application of *model transformations* can be considered an interesting alternative to directly specifying the initial business process model into a specific formal language. To use integration facilities, given by a Model Transformation Language (MTL), has the main advantage of permitting the analyst to continue working with a tool based on a model representation language he is familiar with, in order to build a semi-formal model of a BP, whereas any necessary formal specification of this model can be automatically generated. Thus, we present here the development of the BTRANSFORMER tool, which is based on *MTL ATLAS Transformation Language* (ATL) (Jouault et al., 2008), which allows the user to transform BPMN models into CSP+T ones, in order to ease the automatic verification of BPs that are considered critical to a particular business (Hiles, 2004).

González A., E. Mendoza Morales L., I. Capel M., A. Pérez M., Méndez E. and Domínguez K..
BTRANSFORMER - A Tool for BPMN to CSP+T Transformation.
DOI: 10.5220/0003430003630366
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 363-366
ISBN: 978-989-8425-55-3

2 BACKGROUND

2.1 BPMN

BPMN has emerged as an important open standard graphic notation for modelling and drawing BPs (OMG, 2009). BPMN specifies a single diagram, called *Business Process Diagram* (BPD) (OMG, 2009), which is a flowcharting technique similar to UML activity diagrams. a BP flow is simply modelled by the representation of *events* that occur to start the BP, *activities* and *tasks* carried out, and the outcome of the BP flow. Business decisions and flow branching are modelled using *gateways*. Furthermore, an activity in the flow can be a sub–process, which can be graphically shown by another BPD connected via a hyperlink to a process symbol. If an activity is not structured into sub–processes, it is considered a task. Tasks are the atomic parts of a BP. A *pool* typically represents an organization or business entity and a *lane,* a department or a business worker within that organization or it may be other things like functions, applications, and systems. BPM activity is understood as a successive refinement process, i.e., once the initial model is obtained, pools can be further partitioned into lanes. Both pools and lanes are to represent BP Participants (OMG, 2009), from which process flows that perform activities and tasks follow.

2.2 CSP+T

CSP+T (Žic, 1994) is a specification language, which extends CSP (Roscoe, 2005) to allow the description of complex event timing sinside single sequential processes. CSP+T is of much use in the behavioural specification of concurrent systems that undergo temporal constrained discrete events. CSP+T operators related with timing and enabling–intervals are the following ones: (a) the special process instantiation event denoted ★ (star); (b) the time capture operator (><) associated to the time stamp function $a_e = s(e)$ that allows storing in a variable a_e (marker variable) the time at which event e (marker event) occurs; and (c) the event–enabling interval $\mathrm{I}(T,t_1).a$, which represents temporal refinements of the untimed system behaviour and facilitates the specification and proof of temporal system properties (Žic, 1994).

2.3 BPMN to CSP+T Transformation Rules

In a previous work (Mendoza et al., 2011), we have extended the semantic proposed in (Wong and Gibbons, 2009) through incorporating CSP+T operators, which allows the definition of a timed semantics for a subset of BPMN entities. This opens up the possibility of using state-of-the-art Model-Checking (MC) tools to check temporal constraints between separate event occurrences. Therefore, the time-capture operator and the event enabling interval construct of CSP+T are used to give a syntactical interpretation to tasks' response times modelling. Thereby, the temporal constraints required by business participants in a BP can be formally specified. Consequently, a more precise and complete timed semantics of BPMN entities is obtained than the one given by local and global BPMN diagrams that represent business collaboration.

3 BTRANSFORMER TOOL DEVELOPMENT

We will cover the most relevant aspects of BTRANSFORMER design.

3.1 Inception Phase

The stakeholders' needs have been addressed with the development of BTRANSFORMER tool; i.e., a tool that allows an analyst to generate a CSP+T formal specification from a semi-formal BPMN model. Special notations with regard to timing constraints are also considered in the tool. The functional requirements of BTRANSFORMER tool can be expressed as it follows: (FR 1) the tool should generate a CSP+T specification from a BPMN diagram, and (FR 2) the tool should allow the representation of transformation rules of BPMN constructs as defined in (Mendoza et al., 2011). As non-functional requirements (i.e., the additional characteristics of the tool) or (NFR 1), we can mention: communication and integration facilities with other tools, and (NFR 2) tool portability. In Figure 1 is shown the Use Cases (UC) diagram modelled to cover the previously defined FRs.

As it can be seen in Figure 1, the *UC Generate* transformation is highlighted. This is because the prioritization of UCs decided by BTRANSFORMER tool stakeholders UC is considered the most architecturally significant for the tool can fulfil its objectives. Thus, the version presented in this work addresses the full development of this UC, and thus covers the satisfaction of FR1. In future iterations of the development the remaining UCs will be developed.

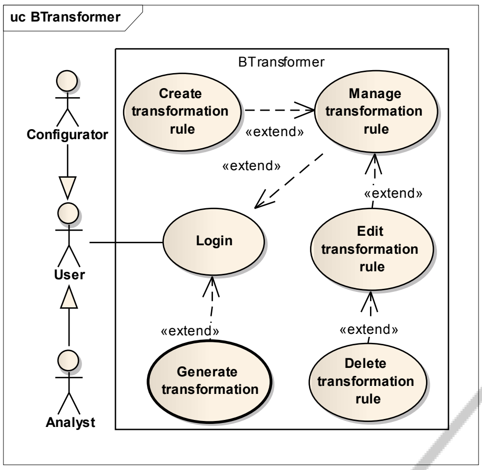

Figure 1: UCs diagram for BTRANSFORMER tool.

3.2 Elaboration Phase

The architectural views suggested at this phase are the following ones: UC view (Figure 1), Logical view (Figures 2), Deployment view, Implementation view and Data view. Due to space limitations, we will only show the most relevant ones.

We present the Class diagram in Figure 2, which shows the main abstractions of BTRANSFORMER tool's design; most of them are suggested by the UC *"Generate Transformation"* and FR 1. The classes implemented during construction of the plug-ins of the transformation tool are defined in Figure 2 and programmed under Eclipse platform (http://www.eclipse.org/), as the main development and implementation environment. Thus, by doing so, we meet the NFR 2 related to portability of the tool.

The class BPMN2CSPTAction represents the actions in the menu that allows accessing the functionality of the tool from Eclipse environment. Class `ApplyTransformationAction` makes use of the services of the environment and is based on the values of one or more classes of `Configuration`. In the class Latex, a format is given to the CSP+T syntactical process terms that result from the transformation and temporal annotations are included. The class `ReaderWriter` allows managing the persistence through the storage of the generated instructions in a file.

The integration of the ATL component in the Eclipse platform is achieved with the editor Intalio (http://www.intalio.com/). Figure 3 displays the components diagram that constitutes the architecture of Eclipse (http://www.eclipse.org/) platform into which the development of the plug-in for BTRANSFORMER is based.

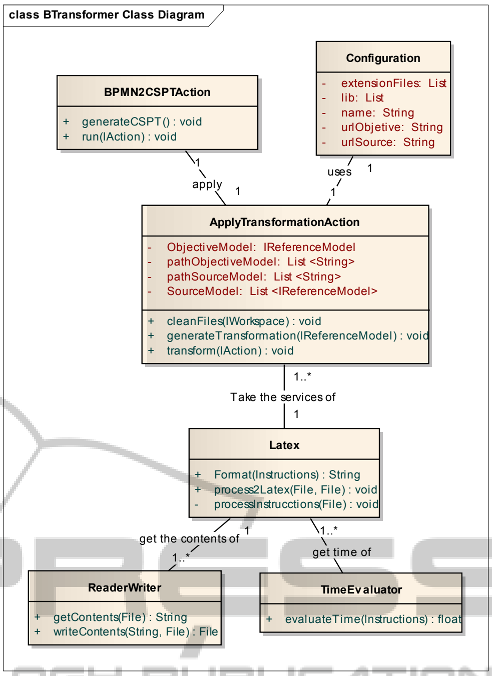

Figure 2: Class diagram of BTRANSFORMER tool.

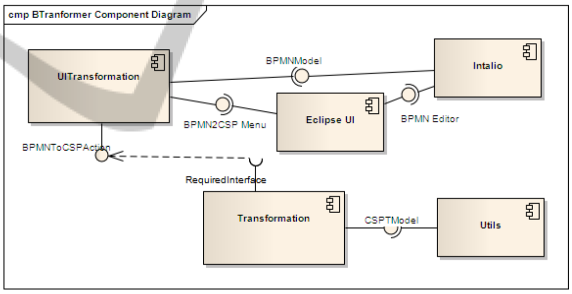

Figure 3: BTRANSFORMER tool components.

The feasibility of the project was increased due to the decision made to use Eclipse platform for the development of BTRANSFORMER. Since processing services of the BPMN editor are always available, it is not necessary to perform the processing of the BPMN model at once. Through the action provided by the Eclipse platform that invokes the plug-in `UITransformationPlugin` (see Figure 3), the access to the functionality of the tool BTRANSFORMER is implemented. The transformation process is facilitated by the plug-in Intalio, which controls the entire BPMN modelling process of the source model, and to which the processing rules defined in the plug-in `TransformationPlugin` are applied to. In its part, the plug-in `UtilsPlugin` handles BPMN modelling entities reading into the source model and writing CSP+T processes to the object model. All plug-ins are based on the platform, however Figure 3 includes, in addition to the plug-ins developed, these that allow the implementation of its interface.

3.3 Construction Phase

All the components and the proposed UC are implemented in this phase. In addition, test cases are applied to each use case. The main deliverable of these phases is the first version of BTRANSFORMER tool, including the source code of all implemented classes. Figure 4 shows the screenshot that corresponds to the invocation of BTRANSFORMER tool according to the sub-option Bpmn2CspT, which activates the plug-in `UITransformationPlugin` of Transformations option. This option can be found in the context menu (right button of the mouse) on the XML file of the BPMN diagram generated by Intalio.

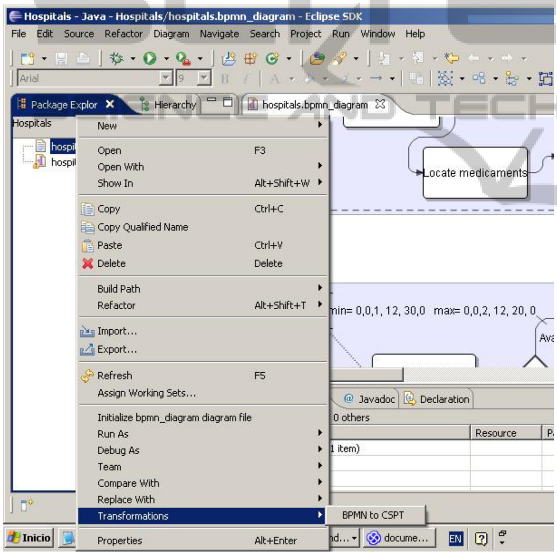

Figure 4: Screenshot of BTRANSFORMER tool invocation.

3.4 Transition Phase

The current installation of the tool represents BTRANSFORMER's implementation on the Information Systems Research Lab department server at Simón Bolívar University, Venezuela, i.e., its final certification/pre-production environment.

4 CONCLUSIONS

This paper introduced the development of BTRANSFORMER tool that allows executing a transformation from a BPMN model into a set of CSP+T process terms. Thus, with our tool a BP analyst is provided with a formal specification of a diagram on which a set of properties are automatically checked. Among the advantages of using this tool, the following ones can be mentioned, (1) it facilitates the generation of specifications in CSP+T from a BPMN model in an automatic way, and (2) does not require the use of external elements to the BPMN standard, in order to specify time management aspects. As future work in the short term we plan to extend BTRANSFORMER and obtain the integration with FDR2. In the long term our aim is to extend the application of the tool to other BP modelling languages and some other PC-based formal specification notations.

REFERENCES

Aalst, W. v . d. and Hee, K. M. v. (2004). Workflow management: Models, methods, and systems. Cooperative information systems. Cambridge, Mass. MIT Press.

Formal Systems (Europe) Ltd, (2005). Failures–Divergence Refinement – FDR2 User Manual. Formal Systems (Europe) Ltd, Oxford.

Hiles, A. (2004). *Business continuity: Best practices : world-class business continuity management. The Rothstein catalog on disaster recovery*. Brookfield, Conn: Roth-stein Associates.

Hoare, C. (1985). *Communicating Sequential Processes*. International Series in Computer Science. Prentice–Hall International Ltd., Hertfordshire UK

Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I. (2008). Atl: A model transformation tool. *Science of Computer Programming 72(1–2),* 31–39.

OMG. (2009). Business Process Modeling Notation – version 1.2. Object Management Group, Massachusetts, USA

Roscoe, A. W. (2005). *The theory and practice of concurrency*. Prentice-Hall international series in computer science. London: Prentice Hall.

Mendoza, L., Capel, M., Pérez, M. (2011) A Formalization Proposal of Timed BPMN for Compositional Verification of Business Processes, *LNBIP* 73, 388 - 403

Weske, M. (2007). *Business process management: Concepts, languages, architectures. Business Process Management.* Berlin: Springer.

Wong, P., Gibbons, J. (2009) A relative timed semantics for BPMN. *Electron. Notes Theor. Comput. Sci. 229(2).*

Žic, J. (1994) Time-constrained buffer specifications in CSP+T and Timed CSP, ACM Trans. Program. Lang. Syst. 16, no. 6, 1661–1674.