Choreography Conformace Checking based on Process Algebras

Manuel I. Capel
Software Engineering Department, University of Granada, 18071 Granada, Spain

Keywords: Business Process Models, Choreography, BPMN 2.0, Transformation Sules Set.

Abstract: Business Process Modelling (BPM) is a *conceptual activity* for representing processes of an enterprise. A *business process* can be understood as a set of related, structured, interacting *activities* driven by a *choreography* that is capable of giving complex services to customers. The ways in which a choreography can be specified in order to check if the behaviour described is conformant with the peer–based interactions of a distributed target system, and thus to prove safety properties, should be correctly defined by identifying the appropriate processes. Hence, choreographies becomes now of great use to analyze and improve any modern company's business.

1 INTRODUCTION

This work is mainly intended to propose a formal semantics to a set of BPMN modelling constructs, useful to enforce the *choreography* to be *realizable*; and then, it introduces an easy approach to the verification of business processes that may use model–checking techniques. Several authors have made progress in solving the problem of BPMN validation. The research work up until now can be divided into two categories, the first one focuses on business process model analysis, whereas the second centres on obtaining a formal semantics of BPMN modelling entities. The main approaches of work in the first area can be found in (Aalst, 2009). In this second group, the central problem to tackle consists of proving the soundness of BPMN model transformation. In many cases, a model cannot be verified because its representation in a executable environment does not show the same *behaviour*[1] as observed in the original model.

The lack of verification exhibited by the aforementioned approaches, in our opinion, it is mainly due to semantic and syntactic problems caused by the incorrect integration of (1) business properties formalization and (2) the corresponding task model.

Taking steps towards the verifiability of BPMN is that we placed both, the executable task model and the BPMN conceptual model, in the same semantic domain. We therefore propose here, (a) a set of transformation rules from a subset of temporal analysis constructs of BPMN into Communicating Sequential Processes + Time (CSP+T) calculus, (b) an easy verification approach for tasks models designed with BPMN. Differently to other authors (Rozinat and Aalst, 2006; Dongen and Aalst, 2004), our method intends to merge the verification process of BP properties with the design of the task model. In this way we take full advantage of the strengths that a formalization of behavioural and temporal aspects of BPMN models can provide to the analysis, both at design and run-time.

2 BPMN CHOREOGRAPHIES FORMALIZATION

BPMN is a standard for the semi–formal specification of task workflows in business BP models and to describe the collaboration between services. BPMN has been extended to BPMN 2.0 to support the collaboration between analysis entities in BP models, which brings forward a *choreographic* model based on peer interactions (Qiu, 2007), instead of following a design model based on services orchestration (Peltz, 2003).

BPMN 2.0 promotes a collaborative and abstract description of software systems that allows for focus-

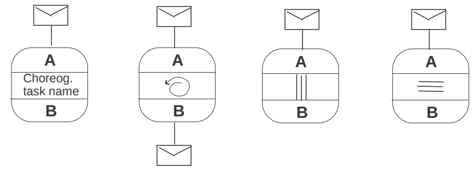

Figure 1: Choreography diagrams.

[1] e.g.,it reacts differently to external events

I. Capel M..
Choreography Conformace Checking based on Process Algebras.
DOI: 10.5220/0004379001350139
In *Proceedings of the 3rd International Conference on Cloud Computing and Services Science* (CLOSER-2013), pages 135-139
ISBN: 978-989-8565-52-5

ing more on what services do in a composition than on how they do it. As consequence, interactions between system's components or *peers* should be more precisely described than in "interconnected interface models", within which the interactions are defined internally to each peer and the description of interfaces between system components becomes of paramount importance. On the other hand, "interaction–based models" consider the "conversations" between peers as the basic building blocks of any system design, whereas the specification of interfaces then becomes secondary to the system's properties analysis.

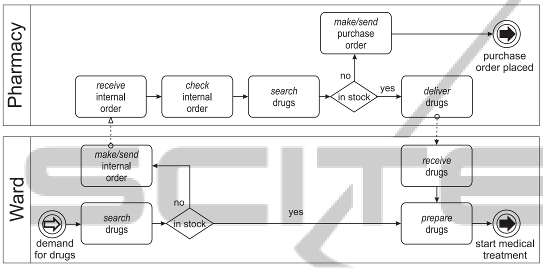

Figure 2: Example of a BPMN diagram.

BPMN 2.0 advocates for the Service Choreography Description Language (WS–CDL), which better fulfills the requirements of choreography specifications due to their global perspective of the system. BPMN 2.0 Choreography Diagrams describe one–way and two–way interactions between peers. In Figure 1, the peers A and B represented by the upper and lower bands, respectively, in the participants tasks exchange messages. In the second task, there is a two way interaction: peer A receives a message and peer B sends a return message.

2.1 Example

Consider, for instance, the simple example of an Hospital Pharmacy logistic process shown in Figure 2. The BPD depicts the message flows between two participants, the *Ward* and the *Pharmacy*, which are independent BP and may have been constructed separately. Clearly, the *synchronization* between both participants is a necessary behavioural property for successful collaboration.

For example, from the pharmacy participant's perspective, drugs delivering should be guaranteed by sending a message to the ward participant prior a purchase order would be made or sent; while from the ward participant's perspective, by sending a message to the pharmacy participant with enough time in advance to have the required drugs to begin medical treatment. The specification in Figure 3 shows the interaction between the peers(`customer`, `ward`, `pharmacy`, `db`) translated from the business process Hospital Pharmacy diagram in Figure 2, which represent the behaviour of tasks within the pools *Ward* and *Pharmacy*. In this specification, we can see that first the customer interacts with the `ward` (`connect`), then sends the prescription to the `pharmacy` (`request`) and eventually receives a response if the drug is available. On the contrary, the `pharmacy` makes a order (`purchase order`) and when the drug is available, it delivers the drug to the `ward` (`receives`). In this last case, the prescription will be prepared and given to the customer (`delivered`), which terminates the complete protocol.

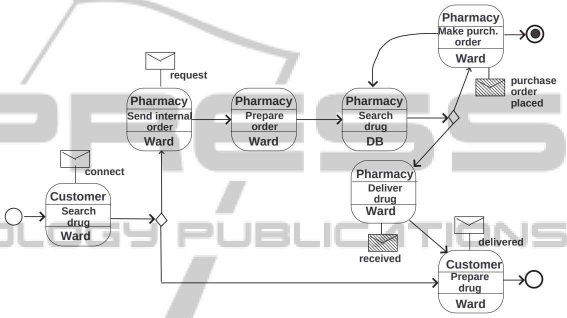

Figure 3: BPMN 2.9 Choreography Diagram–Ward-Pharmacy Example.

2.2 CSP+T

CSP+T (Zic, 1994) is a real–time specification language which extends *Communicating Sequential Processes* (CSP) allowing the description of complex event timings, within a single sequential process, for use in the behavioural specification of any critical communicating process. A CSP+T process term $\mathcal{P}$ is defined as a tuple $(\alpha P, P)$, where $\alpha P = Comm\_act(P) \cup Interface(P)$ is the *communication alphabet* of P.

CSP+T is a superset of CSP, the latter being changed by the fact that traces of events become *pairs* denoted as $t.a$, where t is the time at which event a is observed.

The event enabling interval $\mathrm{I}(T,t_a) = \{t \in \mathcal{T} \mid rel(t_a,v) \leq t \leq rel(t_a+T,v)\}$ indicates the time span where any event is accepted. $rel(x,v) = x + v - t_0$, t_0 corresponds to the preceding *instantiation event* (★), occurred at some absolute time t_0, and x is the value held in the *marker variable* v at that time. The time interval expression can be simplified to $\mathrm{I}(T,t_a) = [t_a, t_a + T]$ if the instantiation event, after which the event a can occur, corresponds to the origin ($t_0 = 0$) of the rt-clock.

3 REIFICATION OF CHOREOGRAPHIES

Any choreography can be considered as realizable if all the interactions that we have specified in the BPMN 2.0 diagram are equivalent to those that can be executed by the interacting peers when we implement the model in a service–description language, such as WS-CDL.

BPMN choreographies are transformed into CSP+T process calculus by following the state-machine behavioural pattern. Each analysis entity of BPMN is encoded as a CSP+T syntactical–process term that represents the machine's state, thereby translating the BPMN construct plus a call to the process that represents the successor state.

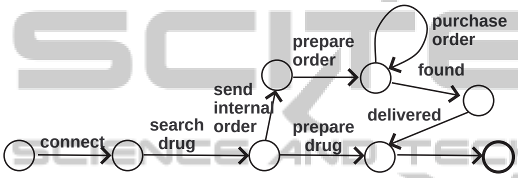

Figure 4: LTS model of the Ward–Pharmacy Example.

The LTS model *reifies* the choreography and allows the verifier to check its realizability w.r.t. the model of the system composed of interacting peers. If these two models are *behaviorally* equivalent, it means that the peer generation exactly satisfies the BPMN communication requirements. On the contrary, the peers do not generate the same interactions as the ones specified in the choreography, and thus we can say that this choreography reification is unrealizable.

The transformation of CSP+T process terms into a LTS model sets the ground for performing behavioural verification of constituent components of a system specified in CSP+T by using the FDR2 MC tool (Formal Systems Europe, 2005). FDR2 checks the behavioural equivalence of two models written as CSP process terms through a refinement relationship between syntactical process terms (Mendoza, 2011).

3.1 Behavioural Equivalence

Taking steps towards the verifiability of BPMN choreographies is that we placed both, the choreography reification and the interacting peers model, in the same semantic domain, that of CSP+T process calculus. We therefore propose here, (a) a set of transformation rules from a subset of analysis constructs of BPMN into Communicating Sequential Processes + Time (CSP+T) calculus, (b) an easy verification approach for choreography models designed with BPMN. Differently to other authors (Dongen and Aalst, 2004; Rozinat and Aalst, 2006), our method intends to merge the verification process of BP properties with the design of the BPMN 2.0 model. In this way we take full advantage of the strengths that a formalization of behavioural aspects of BPMN models can provide to the analysis, both at design and run–time.

Thus, our method to check the realizability of a given choreography reification consists of the following integrated steps according to MC technique and the automata theory,

1. We generate the choreography reification as a LTS derived from the CSP+T encoding.
2. The peers' behaviour are extracted from of the initial BPMN–CD by the application of a proposed transformation rules set.
3. We build the distributed system model, structured as the extracted interacting peers.
4. Finally, we automatically and compositionally (see Theorem 1) verify that the choreography reification (LTS) and the distributed version of the system are behaviorally equivalent.

Theorem 1. ***System Compositional Verification.*** *Let the choreography reification* $\mathbb{C}$ *be structured into several components working in parallel,* $\mathbb{C} = \|_{i:1..n} C_i$. *For a set of* $LTS(C_i)$ *describing the behaviour of components* C_i*, properties* ϕ_i*, invariants* ψ_i*, and deadlock* δ*, with* $\bigcap_{i:1..n}\Sigma_i = \emptyset$, $\bigcap_{i:1..n}\Omega_i = \emptyset$*, and* $\bigcap_{i:1..n}\mathcal{L}(TBA(C_i)) = \emptyset$*, the following condition holds:*

$$LTS(\mathbb{C}) \vDash (\phi\wedge\psi\wedge\neg\delta) \Leftrightarrow \mathop{\Big\|}_{i:1..n} LTS(C_i) \vDash \bigwedge_{i:1..n}(\phi_i\wedge\psi_i)\wedge\neg\delta, \qquad (1)$$

where $LTS(\mathbb{C}) = \|_{i:1..n} LTS(C_i)$.

The practical application of relation (1) includes performing an inductive 'satisfaction checking' process on the range of the components number ($i : 1,\ldots,n$) of the system of peers to be checked .

3.2 BPMN to CSP+T Transformation

We need the semantic precision given by a formal language to the basic analysis entities in order to be able to correctly describe fully executable CDs, such as the one shown in Figure 1. BPMN 2.0 defines advanced constructs, such as different types of OR decision gateways, multiple instances of tasks and subprocesses, which may be transformed into CSP+T process terms, preserving the interaction behaviour between the participant tasks.

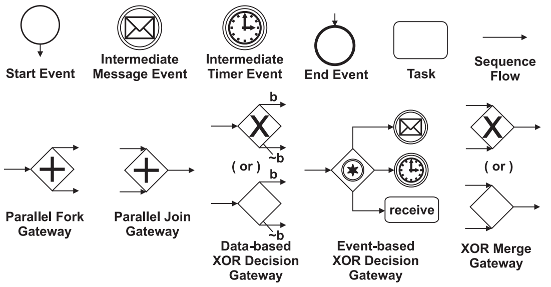

Figure 5: BPMN Elements Extended Graphical Representation.

3.3 BPMN Temporal Extension

In many models of choreographies, constraints on time and resources appear that may cause the violation of the system's safety properties. It becomes therefore necessary to extend the formal notation with new modelling entities that address temporal and resource constraints (Figure 5). We opted to extend the activity symbol with maximum and minimum time instants, within which any task or subprocess execution must be performed.

4 CHOREOGRAPHY MODEL CHECKING

As an application of the formal semantics proposed in section 3.2 for BPMN 2.0 analysis entities, we opted to estudy the *Ward–Pharmacy* example, whose BPMN-CD (Figure 3) shows a certain interaction between the participant peers. For each participant in the BPMN-CD, we specified a parallel composition of parallel CSP+T processes, thereby we can define a bijection between processes and diagram states. The proposed formal specification abstracts the internal interaction between the individual peer states.

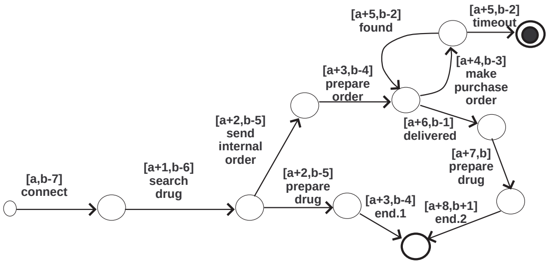

Figure 6: Timed LTS model of the Ward-Pharmacy.

4.1 Verification

The reification of the choreography is realizable if the set of interactions specified by the above process term $T(LTS)$ and those executed by the interacting peers in the target distributed system, specified by the process $P_{BPMN} = Customer \parallel Ward \| bpmn \parallel DB$ are the same. Thus, according to *traces and failures* semantics of CSP, it must be ascertained that the following refining assertion is true,

$$T(LTS) \sqsubseteq_F P_{BPMN} \qquad (2)$$

However, the FDR2 returned *false* since the trace $<$ `connect`, `searchdrug`, `sendorder`, `prepareorder`, `deliver`, `preparedrug` $>$ appears in both models, but the trace $<$ `connect`, `search drug`, *send* `order`, `prepare order`, `deliver`, `prepare drug`, `purchaseorder`, `abort` $>$ is present in the peers–based distributed system and not in the LTS of the choreography.

The solution to this error in the LTS model is to make explicit an extra state in which the LTS is waiting for completing the purchase of the drug and add a timeout to this state If that time period expires then the LTS will reach an `abort` state signifying that the purchase is cancelled, since probably the distributor's stock has been exhausted. The new LTS can be seen in Figure 6.

5 CONCLUSIONS

To check the behavioural equivalence between an LTS, which specifies the possible behaviour of a highly interactive system, against the actual behaviour of a peer–based distributed applicationis is a very complex activity. We have solved that problem by transformation of the LTS and the peer–system model into a proces calculus named CSP+T. In this way, we can analyze and automatically verify the conformance of the defined choreography for services that communicate through messages in a general, distributed, and highly parallel system.

REFERENCES

Aalst, W. (2009). Challenges in business process analysis. in: Enterprise information systems. In *Lecture Notes in Business Information Processing, v. 12:27–42*. Springer–Verlag.

Dongen, B. and Aalst, W. (2004). *Multi–phase process mining: Building instance graphs*. Springer-Verlag, Heidelberg(D), lecture notes in computer science, v. 3288 edition.

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

Mendoza, L. (2011). *Una Contribución a las Técnicas Avanzadas de Verificación de Procesos de Negocio (In Spanish). PhD Dissertation book*. University of Granada (ISBN:978-980-12-4957-3).

Mendoza, L., Capel, M. I., and Pérez, M. (2012). Conceptual framework for business processes compositional verifications. *Information and Software Technology*, 54(2):149:161.

Peltz, C. (2003). Web services orchestration and choreography. *IEEE Computer*, 36(10):46–52.

Qiu, Z. e. a. (2007). Towards the theoretical foundation of choreography. *Proceedings of the 16th international conference on World Wide Web (WWW'07)*, pages 973–982.

Rozinat, A. and Aalst, W. (2006). Springer-Verlag.

Schneider, S. (2000). *Concurrent and Real–Time Systems – The CSP Approach*. John Wiley & Sons, Ltd.

Zic, J. (1994). Time–constrained buffer specifications in CSP+T and Timed CSP. *ACM TOPLAS*, 16(6):1661:1674.