Towards the Integration of Ontologies with Service Choreographies

Mario Cortes-Cornax, Ioana Ciuciu, Sophie Dupuy-Chessa, Dominique Rieu, and Agnès Front

Laboratoire Informatique de Grenoble – Univ. Grenoble-Alpes
{Mario.Cortes-Cornax,Ioana.Ciuciu,Sophie.Dupuy,Dominique.Rieu, Agnes.Front}@imag.fr

Abstract. This paper discusses the integration of ontologies with service choreographies in view of recommending interest points to the modeler for model improvement. The concept is based on an ontology of recommendations (evaluated by metrics) attached to the elements of the model. The ontology and an associated knowledge base are used in order to extract correct recommendations (specified as textual annotations attached to the model) and present them to the modeler. Recommendations may result in model improvements. The recommendations rely on similarity measures between the captured modeler design intention and the knowledge stored in the ontology and knowledge bases.

Keywords: business process, service choreography, ontology, knowledge base, annotation, recommender system.

1 Context and Motivation

Today, organizations are moving towards inter-organizational communications. Therefore, their business processes depend on services provided by external organizations. Modular solutions relying on service oriented approaches (Service Oriented Computing - SOC) [1] are becoming popular in the implementation of business processes where business activities are conceived as services. One way to realize processes involving services is by means of **service choreography** [2], which gives an overall view of interactions between organizations. The global view provided by choreographies is necessary to better understand, build, analyze and optimize inter-organizational processes. This interest is illustrated within the CHOREOS project [3]. One of the applications of this project is to achieve a "passenger-friendly airport" that coordinates the different airport services in a decentralized way.

However, **analyzing and optimizing is not a one shot process**. They need to be integrated into a continuous improvement cycle [4]. In our vision, business processes are modeled in languages such as BPMN [5]. Then, they are annotated by quality recommendations such as average latency or the number of messages sent and received. For that purpose, we plan to use ontologies.

Ontologies are used in this paper in order to model the domain knowledge related to the recommendation annotations attached to the choreography elements. Ontology

Y.T. Demey and H. Panetto (Eds.): OTM 2013 Workshops, LNCS 8186, pp. 343–352, 2013.

is a formal way to represent a common, explicit, agreed-upon conceptualization of a domain in view of interoperability [6]. Choreographies are highly-collaborative models, which imply the need for (semantic) interoperability due to heterogeneity of the background knowledge of each participant. Our main motivation for coupling ontologies with the choreography model is in order to enable reasoning in view of information extraction. The objective of this study is to enrich the choreography model proposed with semantics by means of ontologies in view of their evaluation and improvement. This paper presents the methodology and an integrated tool to help modelers understand, better exploit and iteratively improve their models. The proposed tool is a recommender system extending an existing tool [7], which captures the process modeler design intention via an intuitive interface. This system relies on ontologies in order to extract recommendations, resulting in well structured semantic annotations. The annotations represent interest points in the model that will guide the extraction of the information concerning execution properties but also model-specific properties.

The structure of the paper is as follows. A motivating scenario inspired from one of the CHOREOS use-cases is presented in Section 2. Section 3 illustrates the overview of the approach. Section 4 and 5 briefly introduce the choreography model and the ontology concepts respectively, in the context of this study. Section 6 relies on the motivating scenario in order to demonstrate the application of our approach. Section 7 discusses related work of the paper. Finally, Section 8 concludes the paper and situates the results within a broader perspective.

2 Motivating Scenario – Unexpected Airplane Arrival

Throughout this paper, we use a case study that we describe hereafter. The chosen case study deals with the management of an unexpected arrival of passengers to an alternative airport because of bad weather conditions at destination. The scenario illustrated in Fig. 1 is inspired from one of the case studies discussed in the CHOREOS project.

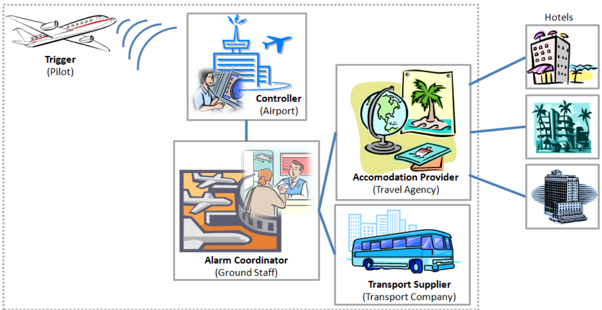

Fig. 1. Illustration of a choreography use case – Unexpected arrival of a plane at an airport

The roles required to implement this choreography are: a *Trigger* which will be played by the pilot, a *Controller* played by the airport, an *Alarm Coordinator* played by the company ground staff, an *Accommodation Provider*, played by a travel agency and a *Transport Provider* which is played by a transport company. In this scenario, we observe that several roles are involved. None of them has the capacity to make the overall decisions as a central control point. Moreover, it would be very risky to have a single point of failure. The solution to this problem is naturally provided by the **choreography** paradigm, which has a distributed nature.

When choreography is modeled, some information, e.g., the load between roles, may be critical. We observe that the load between roles could be measured using different metrics such as the number of relations, the average latency, the number of messages sent and received, etc. This information could be extracted when the choreography is enacted, but also the structure of the model could give some insights about the balance (e.g., the number of relations). Also, a modeler might be interested in particular parts of the model because they are considered critical (e.g., the relationship between *Controller* and *Alarm Coordinator)* whereas other parts might be considered to be less important. The different nature of the information and the need of personalization motivate the inclusion of structured annotations as survey points in the model. The annotations will help extracting valuable information in order to better understand and improve the choreography model. An example of annotation that a modeler may be interested to insert in a choreography model is given in section 5.1.

3 Approach Overview

This section presents the global overview of the approach. We followed a continuous improvement process to describe the integration between ontologies and service choreographies. The steps in the methodology are derived from Deming's PDCA cycle (*Plan Do Check Act*) [4] (See Fig. 2). This process is modeled using the BPMN 2.0 process notation [5]. In addition to the methodology phase, this model also represents the different states in which the choreography model will evolve as well as the information bases related to the ontology.

The phases concerning the methodology are the following: (i) **Model**: this phase refers to the construction of the requirements elicitation and the construction of the choreography model. (ii) **Annotate**: this phase refers to the enrichment of the choreography model with the recommendation annotations. The modeler indicates here what the interest points within the model are. The modeler relies on a knowledge-based (recommendation) system as an annotation helper, on recommendation ontology and on a knowledge base that stores correct recommendations. (iii) **Survey**: in this phase, the choreography is enacted and then studied. The numerical values corresponding to the recommendation annotations will be extracted. (iv) **Improvement Analysis**: in this phase the analysis of the evaluated model is realized. The modeler might decide what corrective actions may be necessary to perform. If the modeler decides that there are no more improvements to do, the process stops. However, if after the analysis, the modeler considers that the choreography model has to be improved, then she returns to the Model phase. If the modeler considers that the annotations are not appropriate to her intentions, she returns to the Annotate phase. (v) **Update**: In this phase, the ontology engineer updates the knowledge base and/or the recommendation ontology with the newly identified recommendations. The annotation system therefore will be limited by the information stored in the knowledge base. This information will be enriched as the system is used and new recommendations are proposed by the modeler and the ontology expert.

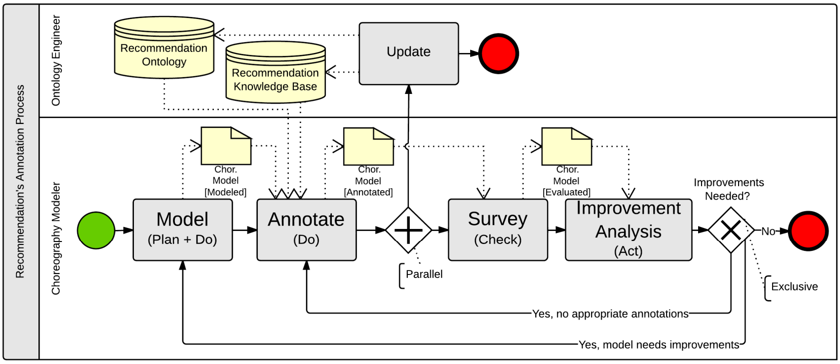

Fig. 2. Overview of the approach

Note that *Choreography Modeler* may represent several organizational roles such as analyst, architect or developer but we did not represent them for simplicity reasons. We focus on the distinction between the parts of the process concerning the choreography model and the ones concerning the ontology. The two parts are explained in the following two sections.

4 Service Choreographies

In this section we introduce the concept of service choreographies through a simplification of choreographies in BPMN 2.0.

4.1 Choreography Model

Choreography generally refers to a description of **coordinated interactions between two or more parties**. Messages exchanged between the different services are tracked from a global viewpoint. There is no central coordinator so all parties are distributed [2]. Decker et al. [8] give a critical overview of different choreography languages such as *Let's Dance*, *BPEL4Chor* and *WS-CDL*. These different proposals seem to converge in the latest version of the Business Process Model and Notation (BPMN 2.0) [5], which is the de-facto standard for business process models. In the latest version of BPMN, choreography has been adopted as a first-class citizen.

In this paper we will only focus on an abstract level of choreographies where the fundamental elements of choreography are presented. This level, illustrated in Fig. 3, might be considered as **a simplification of choreographies in BPMN 2.0**. The small differences in this level from BPMN 2.0 are justified by several issues detected when evaluating the support of choreographies in BPMN 2.0 [9].

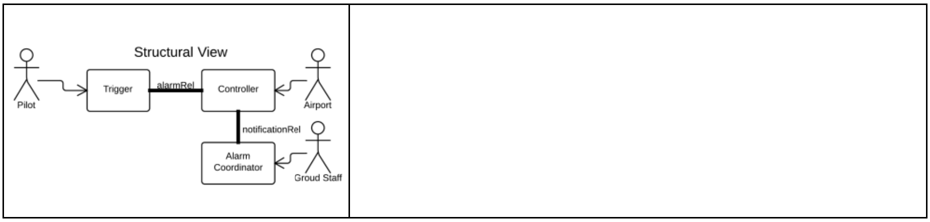

Fig. 3. BPMN 2.0 simplification described by a Structural View and a Behavioral View

The choreography model in Fig. 3 is divided in two views: (1) the structural view ; and (2) the behavioral view. In the **structural view**, let us consider a set of **roles** (*Role*) as for example *Trigger* and *Controller* linked in two by two **relationships** (*RoleRelationship*). The relationship represents the existence of a previous knowledge between both roles. A role is an abstract entity played by an **actor** (*Actor*). For example, the *Pilot* plays the *Trigger* role. In the **behavioral view,** we represent the flow of interactions between roles.

After explaining the choreography concept, let's look in more detail at the kind of information that we are interested to extract and how this is represented in the model.

4.2 Choreography Recommendations

Our approach is somehow similar to monitoring approaches [10]. However, process monitoring mainly focus on execution. In our study, we also want to extract useful information from the model construction. For example, if a modeler wants to look at the roles load, some useful information could be directly extracted from the model without execution. Just considering the number of relations defined for a role, or the number of interactions the role is involved in, a modeler may conclude that the model should be improved. This information could be complemented by additional information extracted at run time. In this study, we focus on information retrieved from the model as a first step of the application of our approach.

We called these interest points **recommendations**. In order to identify them, we looked at cross organizational monitoring and QoS requirements [10]. We also looked at choreography modeling requirements [9]. The classification of these requirements is out of the scope of this paper, therefore we do not go into details. Hence, the goal is to provide to modelers a mean to annotate a choreography model so that this markers result in information extraction that will be used to improve the model. The following section explains how these recommendations are defined and structured by means of ontologies. In Section 6 we explain in more detail the extraction of information though an application scenario.

5 Ontologies for Service Choreographies

This section illustrates how the choreography model presented in the previous section is enriched with semantics. It describes 1) the recommendation ontology and knowledge base used to annotate the choreography model; 2) the structure of the recommendations; and 3) the supporting annotation tool proposed.

5.1 Recommendation Ontology and Knowledge Base

In this study, the recommendations attached to the model elements are represented by an OWL ontology designed under Protégé[1].

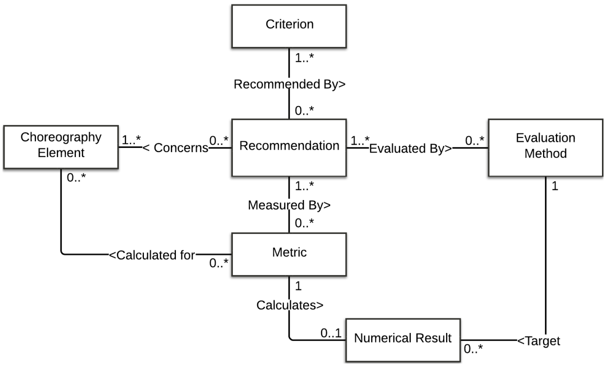

Fig. 4. Overview of the Recommendation Model

The **recommendation ontology** consists of facts and application-specific constraints applied to these facts. An example of a fact is: "a recommendation is measured by a metric / a metric measures a recommendation". A constraint imposed on this fact could be "There is at least one recommendation associated to a metric".

The **knowledge base** stores a set of correctly defined instances of recommendations. The recommendations in the knowledge base represent a collaborative (community) and continuous effort. The recommendations need to be approved by an expert (ontology engineer) before entering the knowledge base. An example of syntactically and semantically correctly defined recommendation that marks a point of interest on a "role" element regarding the number of relationships in order to analyze the choreography load balance among roles is:

`@RoleLinkability [criterium="Load" choreography element ="Role" metric="RoleRelationshipNb"].`

The recommendation ontology and knowledge base are used to annotate the following elements of the choreography model: *Role*, *Actor*, *RoleRelationship*, *Gateway*, *Choreography Activity*, *Event* and *Choreography*. Fig. 4 gives an overview of the recommendation ontology structure that we propose. This model is inspired from the

[1] http://protege.stanford.edu/

QUIMERA quality meta-model proposed by Frey et al.[11]. A recommendation applies to at least one choreography element (1..*). A recommendation is in relation to a criterion which it recommends (1..*), it is measured by zero or more metrics (0..*) and is evaluated by an evaluation method, according to numerical results returned by the metric(s). The structure of a recommendation annotation within the ontology is described in the following sub-section.

5.2 Recommendation Annotation

A recommendation annotation is a text annotation attached to a choreography element in order to mark a point of interest in the model. The syntax of a recommendation annotation is specified by a recommendation term, followed by a parameter-value list, as follows: `@RecommendationTerm [parameter = value]`.

The parameters are: the criterion, the choreography element being annotated and one or more metrics representing numerical information about the recommendation.

It could be argued that the insertion of many annotations into the choreography model could result in a visually overloaded model. Therefore, we will propose graphical markers to visually represent annotations while managing the complexity of the annotated model.

5.3 Extraction of Recommendations

The specification of the recommendations is not a straightforward task, especially when dealing with tens or hundreds of recommendations, with their intrinsic properties. In order to overcome the difficulties the process modeler might encounter when defining a recommendation annotation, we propose a knowledge-based recommender system as a supporting tool for our methodology. The recommender system is able to: 1) **capture** the process modeler intention thanks to a user-friendly interface; 2) **compare** it with the knowledge expressed in the ontology and in the knowledge base; and 3) **propose** a valid set of recommendations.

The tool is based on a previously proposed solution [7]. The proposed architecture contains several modules encapsulating the main functions of the system. In this study, we focus on the annotator and the retriever components adapted to choreographies: 1) the **annotator** is the module that assists the user with concepts from the ontology base, in order to define syntactically correct annotations. It is used either as an auto-completion operation while the user is typing in the desired elements or as an ontology browser; 2) the **retriever** component extracts objects from the knowledge base that are similar to the input object provided by the user. This component is based on graph-based retrieval (e.g., SPARQL queries) and on various similarity measures. For example, the user only wants to retrieve those objects with a particular value for "criterion". The result of the retrieval operation is a set of recommendations.

The retriever performs similarity check in order to compare and classify the user request with the set of recommendations available in the knowledge base. The **matching** can be performed at a **string**, a **lexical**, or a **graph** (**ontology**) level depending on the user input. For example, when a user performs a typing error, the system could perform a string matching operation in order to find the correct term. In other cases, when several organizations participate in the design of a choreography model, the differences between their vocabularies must be taken into account. The system handles this by performing lexical matching according to predefined organization-specific dictionaries.

The following section shows how the annotation approach was applied in order to define correct recommendation annotations within the *unexpected plane arrival* scenario presented in Section 2.

6 Example of the Application of the Methodology

In this section, we present the application scenario relying on the motivating scenario of Section 2. Fig. 5 shows a simplified choreography model in its three possible states. At the end of the *Model* phase, we get a **Modeled** choreography model. In the *Annotate* phase, the recommendations will be introduced relying on the Recommendation System to produce an **Annotated** model. The structured annotations are extracted relying on user's keywords as explained in Section 5.3.

Fig. 5 shows that we might have several *RoleLinkability* recommendation instances linked to different elements of the model. The example also illustrates how the same recommendation may be linked to different metrics (i.e., different ways to calculate the numerical values). In the *Survey* phase, the annotations may extract some numerical values and some interpretation help so that the modeler can proceed to the analysis. The way these values are calculated has to be defined in the evaluation method (see Fig. 4) attached to a recommendation. For the moment, we focus on

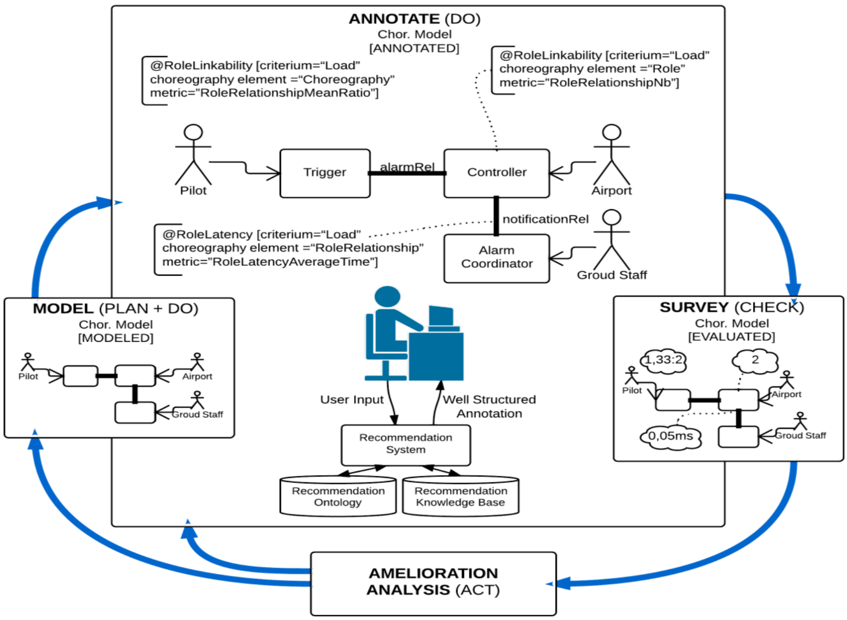

Fig. 5. Overview of the Application Scenario

model properties that can be easily extracted from the model serialization (e.g., the number of relationships). Note that the numerical values given in the figure are just an example and that the extraction of the execution properties is not yet implemented. The resulting model in this phase is **Evaluated**. The latter model is the entry point for the *Improvement Analysis* phase where the modeler decides if the model should be improved and if the defined annotations are appropriate or there may be changed.

This scenario shows the interest of the approach giving an application example that implements the methodology explained before. It also illustrates how ontologies, via the recommender system, are integrated with choreographies.

7 Related Work

The importance of adding semantics to business process models has been recognized within the business process community for several years now [12]. Currently, Semantic Web technologies are applied in order to enrich the business processes. The common instrument for specifying the semantics are the semantic annotations, used in order to either model the process dynamics [13] or the process structure, as demonstrated within the SUPER project [14]. Other approaches focus on the semantic interoperability of the business process knowledge [15]. Moreover, semantic annotations have been added to business processes in order to provide user support during the model design process, as described in [16]. This paper is mostly in-line with the last two mentioned approaches, aiming at capturing the user design intentions and providing appropriate recommendations to the process modeler and in the same time enforce the semantic interoperability.

Regarding the recommender systems, they have been extensively used applications such as, for example, e-commerce [17]. Semantic recommender systems are the next-generation recommender systems [7]. This study goes beyond the state of the art, focusing on personalized recommendations, by capturing the user design intentions and their similarity to an evolving knowledge base and ontology.

We extend our previous work by proposing a user-friendly recommender system which is intended to support the modeler during the evaluation and improvement of a choreography model.

8 Conclusion and Future Work

This paper presents an ongoing study for the retrieval of choreography recommendations in view of model analysis and improvement. The paper presents a methodology that brings together choreography models and ontologies. The paper suggests relying on a continuous improvement process based on these recommendations to help modeler(s) in the design of choreographies.

Future work concerns the extension of the current study with an intuitive interface with respect to capturing the process modeler design intentions and the presentation of the results. We are planning to integrate an existing prototype of a graphical editor for choreographies with the aforementioned ontology-based tool Protégé, so that we can query the ontology from the editor to extract the annotations. Another interesting approach could be to use the annotated criteria for feeding a simulation model.

The extension of the retriever component with new similarity metrics is work in progress. The enrichment of the knowledge base with expert knowledge and the study of automatizing its evolution is also ongoing work.

Another interest point to be studied is the possibility to link the recommendations and the corresponding numerical value to the three levels of the choreography meta-model that we defined in our previous work in view of adapting the extracted information to the desired level of detail. We consider validating the approach by an empirical experiment to observe if recommendations help modelers to improve their models without adding extra-complexity.

References

1. Papazoglou, M.P., Georgakopoulos, D.: Service-Oriented Computing. Comm. of the ACM 46(10), 24–28 (2003)
2. Peltz, C.: WS orchestration and choreography. Computer 36(10), 46–52 (2003)
3. CHOREOS project: http://www.choreos.eu/bin/view/Main/
4. Deming, W.E.: Out of the crisis, vol. 4(1). MIT Press (2000)
5. OMG, Business Process Model and Notation, Business (2011)
6. Gruber, T.R.: Toward Principles for the Design of Ontologies used for Knowledge Sharing. In: Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers (1993)
7. Ciuciu, I., Zhao, G., Mülle, J., von Stackelberg, S., Vasquez, C., Haberecht, T., Meersman, R., Böhm, K.: Semantic support for security-annotated business process models. In: Halpin, T., Nurcan, S., Krogstie, J., Soffer, P., Proper, E., Schmidt, R., Bider, I. (eds.) BPMDS 2011 and EMMSAD 2011. LNBIP, vol. 81, pp. 284–298. Springer, Heidelberg (2011)
8. Decker, G., Kopp, O., Leymann, F., Weske, M.: BPEL4Chor: Extending BPEL for Modeling Choreographies. In: IEEE Int. Conf. on Web Services, pp. 296–303 (2007)
9. Cortes-Cornax, M., Dupuy-Chessa, S., Rieu, D., Dumas, M.: Evaluating choreographies in BPMN 2.0 using an extended quality framework. In: Dijkman, R., Hofstetter, J., Koehler, J. (eds.) BPMN 2011. LNBIP, vol. 95, pp. 103–117. Springer, Heidelberg (2011)
10. Wetzstein, B., Karastoyanova, D., Kopp, O., Leymann, F., Zwink, F.: Cross-organizational process monitoring based on service choreographies. In: Proc. of ACM Symp. on Appl. Computing (2010)
11. García Frey, A., Céret, E., Dupuy-Chessa, S., Calvary, G.: QUIMERA. In: Proc. of 3rd ACM SIGCHI Symp. on Engineering Interactive Computing Systems, pp. 265–270 (2011)
12. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic business process management: a vision towards using semantic Web services for business process management. In: IEEE Int. Conf. on e-Business Engineering, pp. 535–540 (2005)
13. Wetzstein, B., Ma, Z., Filipowska, A., Kaczmarek, M., Bhiri, S., Losada, S., Lopez-Cobo, J.M., Cicurel, L.: Semantic business process management: A lifecycle based requirements analysis. In: Proc. of Semantic Business Process and Product Lifecycle Mgt. (2007)
14. SUPER Project: http://www.ip-super.org/
15. Lin, Y.: Semantic annotation for process models: Facilitating process knowledge management via semantic interoperability, Norwegian University of Technology (2008)
16. Betz, S., Klink, S., Koschmider, A., Oberwiseis, A.: Automatic user support for business process modeling. In: Proc. of Semantics for Business Process Mgt., pp. 1–12 (2006)
17. Schafer, J., Konstan, J., Riedl, J.: E-commerce recommendation applications. In: Data Mining and Knowledge Discovery, vol. 5(1-2), pp. 1–24 (2001)