Expert Systems With Applications 99 (2018) 103–114

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Model visualization: Combining context-based graph and tree representations

Kimberly García*, Patrick Brézillon

Laboratoire LIP6 - Université Pierre et Marie Curie, 4 Place Jussieu, Paris Cedex 5, 75252, France

ARTICLE INFO

Article history:
Received 23 May 2016
Revised 20 January 2018
Accepted 21 January 2018
Available online 31 January 2018

Keywords:
Context modeling
Contextual graph formalism
Tree representation
Contextual information
Task modeling
Decision making

ABSTRACT

A Contextual Graph is based on the Contextual Graph formalism, which allows experts to realistically model the possible ways a task can be realized (i.e., practices). The power of Contextual-Graphs relies on their capability of considering the situation-dependent data (i.e., contextual information) that characterizes a task realization. Through successful experiences applying this formalism in several fields (e.g. medicine, transport, and military) it has been identified a clear need for an alternative representation of tasks that require an immediate response to an event (e.g. an incident in the Parisian subway). Decision makers have expressed their interest in a model that allows them to quickly identify the contextual information needed to elaborate and develop a practice. Moreover, in task realizations in which either a real object (e.g. a medical image), or an object of the reasoning (e.g. a strategy for driving in a freeway) corresponds to a practice output, a graph representation is not the clearest visualization, nor the fastest support for decision makers. The purpose of this paper is to offer a tree view of practices in a Contextual Graph. Such tree representation responds to the decision makers needs, providing another outlet for analyzing a task realization and its outputs. Moreover, the automatic generation of this view was implemented as a feature of the CxG software.

1. Introduction

The concept of context has been widely defined in several fields of study. In 2005 (Bazire & Brézillon, 2005) created a corpus of over 166 definitions of context proposed in several fields of study. In Computer Science, the most popular definitions are the ones from Brézillon and from Dey. In Brézillon and Pomerol (1999) context is defined as “what does not intervene explicitly in a focus but constrains it”, while in Dey, Abowd, and Salber (2001) ”context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves”. By analyzing these two popular definitions, it is not surprising that the research conducted in the context field can be divided into two categories (Vieira, Tedesco, & Salgado, 2011): (a) the one aiming to create context-aware applications (Perera, Zaslavsky, Christen, & Georgakopoulos, 2014), and (b) the one focused on formalizing, modeling, and creating frameworks to apply the context concept to different fields.

In the former category, which is more aligned to Dey’s definition of context, many tools have been used to store and make available to applications data that characterizes the current situation of a user. These tools include tag languages such as XML, used to create hierarchical documents in which contextual data (e.g. location, temperature, and time) captured by sensors can be associated to a user of the application. Such data is then processed by the application to customize its content and/or behavior, providing context-awareness to the user. Other popular tools used by context-aware applications are UML representations and OWL ontologies. The objective of using these tools is to create models to support enriched contextual information, which is not just provided by sensors, but which is inferred by relating such sensor data to instances in the corresponding knowledge model. This is the case of applications created to adjust their interface and functionalities based on the user’s activity in progress, or on the people the user is currently accompanied by.

Concerning the second type of research in context, which is the one we focus on this paper, some formalisms for representing context are Contextual Graphs (Brézillon, 2007), and Description Logics (Klarman & Gutiérrez-Basulto, 2011). Description Logics is an expressive rule-based formalism that has been used to represent contextualized knowledge. However, such formalism does not provide a graphic representation that can be understood by

* Corresponding author.
E-mail addresses: kimberly.garcia@siemens.com, kimberly.garcia@lip6.fr (K. García), patrick.brezillon@lip6.fr (P. Brézillon).

https://doi.org/10.1016/j.eswa.2018.01.033
0957-4174/

non-experts, which makes difficult its incorporation into real world setups. On its side, the Contextual Graph formalism has been conceived to support decision-making and knowledge transfer by providing semantic items to represent task realizations as they occur in real life. Even though, a Contextual Graph (CxG) is a graphic representation, it does not provide an immediate association between a practice and its output, since a CxG is an experience base, in which all the practices (experiences) are integrated as a whole. Thus, a different representation is needed; one that provides a way to communicate what to do in a specific context: the roadmap to make a decision. Therefore, in this paper, we propose the introduction of the Practice Tree view. This view is obtained by transforming a Contextual Graph into a tree, in which the contextual elements found are placed on the left side of such a tree, and the actions that should be followed to complete the task are placed on the right side of the tree.

While a Decision Tree might look similar in structure to a Practice Tree and in some cases considers contextual variables as the decision criteria to predict an outcome (Ghattas, Soffer, & Peleg, 2014). These tools are different in nature, since the latter is derived from a collection of experiences (CxG) that includes the different actions and activities (smaller tasks) that need to be performed in order to complete a task. Since the final user is a human making decisions based on all the entities that involve a way of performing a task, actions and activities are as important as Contextual Elements (decisions). The Practice Tree view is meant to complement a CxG, not to replace it. This view is the response to feedback that users have given in tens of real worlds applications deployments of the CxG software. On its side, a decision tree is focused just on the decision criteria, and the final outcome, but not in the process to get to these decisions. Decision Trees are usually used as a mathematical approach for prediction, and classification (Rokach & Maimon, 2014), rather than a tool for humans.

In the following, we first present a quick overview of the Contextual Graph formalism in Section 2. Then, the theoretical framework for understanding the characteristics of contextual graphs and their transformation into trees are described in Section 3. Following, in Section 4 the inclusion of the Practice Tree view to the CxG software is explained, as well as the difficulties that have been overcome to create such a view, and the new ideas we aim to explore through their actual implementation. Finally, in Section 5 the lesson learned and the future direction of this work are presented.

2. On the contextual Graph formalism

In Brézillon and Pomerol (1999) context is defined as "what does not intervene explicitly in a focus but constrains it". Based on this working definition, the implementation of the Contextual-Graphs (CxG) formalism leads to a uniform representation of elements of knowledge, reasoning and context that are needed in a task realization. Each path in a contextual graph represents a way of realizing the task in a particular situation (i.e., context). Hence, a CxG is a collection of knowledge and experiences that experts accumulate by performing the task in different conditions. Four types of items can be found in a CxG (Brézillon, 2005): actions, contextual elements, activities and Executive Structures of Independent Activities (ESIA). An action is the building block of the modeling at the given granularity of the representation (represented by a green square box) (see Fig. 1). A contextual element (CE) is a pair of nodes, a contextual and a recombination one (represented by a numbered and a not numbered blues circle). The former has one input and N outputs (exclusive branches) corresponding to N known values of the contextual element. The latter is a [N, 1] relationship that represents the moment at which the instantiation of the contextual element does not matter anymore. An activity represents a complex action described as a contextual graph by

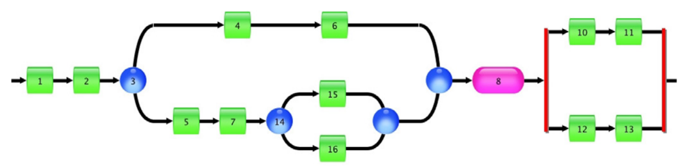

Fig. 1. Example of contextual graph. (For interpretation of the references to colour in the text, the reader is referred to the web version of this article.)

itself, and actors in different task realizations can identified it as a work unit. Finally, an ESIA (represented by enclosing parallel vertical bars) expresses the execution of different independent (sub-)activities in a parallel or sequential way, regardless of the order. To some extent, an ESIA is a complex contextual element.

A contextual element corresponds to an element of the nature that is evaluated when needed. The value taken (instantiation) by a contextual element when the focus is on it is considered as long as the situation is under analysis. The proceduralized context, which corresponds to the ordered sequence of instantiated contextual elements during a practice development, evolves dynamically while the focus progresses. According to previous uses of the CxG in a vast range of applications, we have learnt that: (1) contextual elements come from different heterogeneous sources that can be organized in four categories: user, task, situation and immediate environment, (2) each possible value of a contextual element corresponds to a "best" way of reaching the sub-goal in a specific context, and (3) two actors may instantiate differently a contextual element in the same activity, since they put the emphasis on different contextual elements. As it can be seen, contextual elements play a key role in decision-making.

A contextual graph has a unique input, a unique output and an organization of nodes connected by edges, in which each node can be an action, an activity, a contextual node or a recombination node. A contextual graph takes a series-parallel structure from sets of elements placed one after the other in a series fashion. Contextual elements provide the parallel structure of a graph, since they lead to several different alternatives of completing a task realization. Branches of a contextual element do not correspond to edges in a series-parallel graph (cf. Section 3), because contextual graphs are semantic by nature, thus each of its branches corresponds to a value taken by the contextual element. Since a contextual element is a unit constituted by a contextual node and a recombination one, when having two contextual elements CE1 and CE2 placed in a series fashion, and integrated by contextual nodes CN1 and CN2, and recombination nodes RN1 and RN2 respectively, it is not possible to have a path such as CN1-CN2-RN1-RN2, since the recombination node of the current contextual element should be reached before going to the next contextual element. Thus, the only allowed paths are CN1-RN1, CN2-RN2, and the full path CN1-RN1-CN2-RN2. The same applies for these two contextual elements placed in a parallel manner, in which CE2 is nested in CE1 (i.e., CE2 is on one of CE1 branches). The recombination node of the nested contextual element should be reached first before visiting the recombination node of the upper contextual element. Thus, the only possible paths are CN2-RN2, CN1-RN1, and the full path CN1-CN2-RN2-RN1.

The novelty of the work presented in this paper, realies on providing decision makers with a different graphical representation of a CxG which allows them to: (a) visualize the individual practices that can be followed to complete a task, (b) quickly identify the contextual information needed to follow each practice, and (c) immediate visualization of the output a practice produces. The feedback obtained from previous deployment of the CxG software for different applications, we decided to explore a tree representation. Thus, it was necessary to explore the formal way to trans-