Information Systems 36 (2011) 881–897

Contents lists available at ScienceDirect

Information Systems

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

Human and automatic modularizations of process models to enhance their comprehension

H.A. Reijers [a], J. Mendling [b,*], R.M. Dijkman [a]

[a] Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
[b] Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany

ARTICLE INFO

Article history:
Received 12 June 2010
Received in revised form
29 December 2010
Accepted 9 March 2011
Recommended by: G. Vossen
Available online 21 March 2011

Keywords:
Business Process Modeling
Modularity
Empirical test
Automated discovery

ABSTRACT

Modularization is a widely advocated mechanism to manage a business process model's size and complexity. However, the widespread use of subprocesses in models does not rest on solid evidence for its benefits to enhance their comprehension, nor are the criteria clear how to identify subprocesses. In this paper, we describe an empirical investigation to test the effectiveness of using subprocesses in real-life process models. Our results suggest that subprocesses may foster the understanding of a complex business process model by their "information hiding" quality. Furthermore, we explored different categories of criteria that can be used to automatically derive process fragments that seem suitable to capture as subprocesses. From this exploration, approaches that consider the connectedness of subprocesses seem most attractive to pursue. This insight can be used to develop tool support for the modularization of business process models.

1. Introduction

In the design and production of complex technology, modularity is recognized as a key principle. For example, it has been argued that the computer industry has dramatically increased its rate of innovation by adopting modular design [1]. In contexts such as these, *modularity* is commonly interpreted as the design principle of having a complex system composed from smaller subsystems that can be managed independently, yet function together as a whole [2].

Modularization is also applied in business process models using *subprocesses*. Most popular process modeling techniques support this concept, e.g. UML Activity Diagrams [3], EPCs [4], BPMN [5], and YAWL [6]. Various advantages are attributed to the use of subprocesses in process models, in particular when they grow large. At build-time, subprocesses support a *modeling style* of stepwise task refinement, stimulate *reuse* of process models, and potentially *speed up* the (concurrent) development of the overall process model [7,8]. At run-time, i.e. when a process model is enacted by an automated system, subprocesses allow for *scaling* advantages: Each subprocess, for example, may be executed on a different workflow server [8]. Finally, when a process model is used to facilitate the understanding of complex business processes among various stakeholders, subprocesses are supposed to ease the *understanding* of the model [9,10]. The latter advantage is particularly noteworthy, because in most business applications it *is* the primary purpose of a process model to act as a means of communication [11,12]. This paper will be concerned with this particular advantage of using subprocess in process models, i.e. the enhancement of their comprehension by human readers.

It should be noted that the way in which modularity is currently utilized in modeling practice raises some questions about its actual benefits from the perspective of

* Corresponding author.
E-mail addresses: h.a.reijers@tue.nl (H.A. Reijers), jan.mendling@wiwi.hu-berlin.de (J. Mendling), r.m.dijkman@tue.nl (R.M. Dijkman).

0306-4379/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.is.2011.03.003

human comprehension. First of all, there are no objective criteria to establish the right level of granularity for a subprocess. Accordingly, there is no absolute guideline if a particular subprocess should be on level X or $X+1$ in a model hierarchy [13]. Neither is there a unique way to modularize a process model [13]. As a consequence, modularity is often introduced in an ad hoc fashion. Furthermore, there are clearly drawbacks when the process logic is fragmented across models. In particular, it "becomes confusing, less visible, and tracking [...] paths is tiring" [14] if a subprocess is decomposed in further subprocesses. The fact that the semantic check in ARIS Toolset mainly addresses consistency issues between events in the subprocess and around the refined function illustrates the seriousness of this problem. Finally, even if modularization is useful for maintenance purposes, by making it easier to understand which aspects must be changed, it is questionable whether advantages materialize in practice: Many organizations fail to keep their models up to date [15].

In this paper, our interest is with two research problems. The first problem is that solid indications are missing for benefits of modularization in process models, i.e. the use of subprocesses, to ease their interpretation. Our interest is to discover whether subprocesses can be useful to improve the *understandability* of real-life process models. For this issue, we will build on an empirical investigation of two complex process models from practice, both in modular and "flat" form, and their comprehension by a group of 28 experienced process modelers. The contribution of our work is to provide tangible support for the usefulness of subprocesses in process models. We also provide an insight into the underlying causes for this effect.

The second problem we address is the lack of dedicated approaches to support process modelers with modularizing a given process model into a group of related, understandable subprocesses. We explore three attractive directions for the automated discovery of subprocesses, apply them to a real-life and complex process model, and evaluate the results against the modularization that experienced process modelers provided for the same model. Our contribution in this respect consists of providing concrete indications for the further development of automated discovery algorithms.

In the presentation of our contributions, we will build on some of our earlier work [16]. In comparison with this publication, we significantly extended the presentation and discussion of the experiment that was conducted to investigate the effect of subprocess usage and updated the review of related literature. Beyond that, the use and evaluation of the automated discovery algorithms that is included in the current paper is completely new.

Against this background, the structure of this paper is as follows. In the next section, we will give a broader background for the concept of modularity, in particular with respect to process modeling. In Section 3 we will present the setup of our empirical test along with its results and a discussion. Section 4 presents our proposals for automatic support for subprocess discovery with a corresponding evaluation. Section 5 compares our contribution to related research. Section 6 concludes the paper.

2. Theoretical background

In this section we discuss the theoretical background of our research. In Section 2.1 we present the essential concepts related to modularity in conceptual modeling. Section 2.2 revisits contributions on the modularity of process models. Section 2.3 takes a cognitive research perspective on process model modularity, and derives hypotheses on its costs and benefits.

2.1. Modularity in system design and conceptual modeling

Often, the terms *modularity*, *decomposability*, and *hierarchy* are used interchangeably. However, according to [2], a modular system is not automatically *decomposable* in the sense that the modules can be easily managed independently. After all, it is possible to break a system into modules whose workings remain highly interdependent with the internal workings of other modules. Furthermore, as Parnas points out in his seminal paper on "information hiding", a modular system is not necessarily *hierarchical* [17]. To clarify these notions, consider Fig. 1. In this figure, three abstract modular designs can be seen. In each of these, a module is represented as a rectangle and each arrow represents a "uses" relation between two modules. Design (a) is hierarchical, since the dependencies form a partial ordering. This is, however, not the case for design (b): A cyclic dependency exists between a subset of the modules. Such a design is called non-hierarchical. Furthermore, designs (a) and (b) may well be decomposable, considering the limited number of dependencies between the modules. In contrast, this is less obvious for design (c) with its numerous interdependencies. Note that the hierarchy notion can be mathematically pinned down, where decomposability refers to a qualitative notion. For this paper we consider the general phenomenon of "modularity" as the main subject of interest.

In many settings, "the real issue is normally not to be modular but *how* to be modular" [2]. Modular systems are much more difficult to design than comparable interconnected systems [1]. Beyond that, problems with incomplete or imperfect modularization tend to appear only when the modules come together and work poorly as an integrated whole. It has been argued that many of the

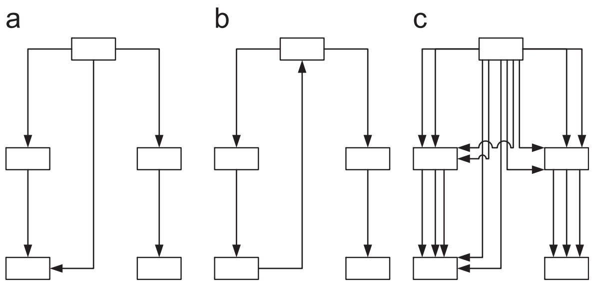

Fig. 1. Examples of modular designs.