Negotiation Among Web Services Using LOTOS/CADP

Gwen Salaün, Andrea Ferrara, and Antonella Chirichiello
DIS - Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italia
Contact salaun@dis.uniroma1.it

Abstract. It is now well-admitted that formal methods are helpful for many issues raised in the web service area. In a previous work, we advocated the use of process algebra to describe, compose and reason on web services at an abstract level. In this paper, we extend this initial proposal, which only dealt with behavioural aspects, to cope with the question of representing data aspects as well. In this context, we show how the expressive process algebra LOTOS (and its toolbox CADP) can be used to tackle this issue. We illustrate the usefulness of our proposal on an important application in e-business: negotiation among web services. The connection between abstract specifications and running web services is made concrete thanks to guidelines enabling one to map LOTOS and the executable language BPEL in both directions.

Keywords: Web Services, Formal Methods, LOTOS, CADP, Negotiation, BPEL.

1 Introduction

Web services (WSs) are distributed and independent pieces of code solving specific tasks which communicate with each other through the exchange of messages. Some issues which are part of on-going research in WSs are to specify them in an adequate, formally defined and expressive enough language, to compose them (automatically), to discover them through the web, to ensure their correctness, etc. Formal methods provide an adequate framework (many specification languages and reasoning tools) to address most of these issues (description, composition, correctness). Different proposals have emerged recently to abstractly describe WSs, most of which are grounded on transition system models [2,11,18,10,14] and to verify WS description to ensure some properties on them [18,17]. In a previous work [19], we advocated the use of process algebra (PA) [3] for WSs. Being simple, abstract and formally defined, PAs make it easier to specify the message exchange between WSs, and to reason on the specified systems (e.g. using bisimulation notions to ensure the correctness of composition).

In this initial proposal [19], we especially experimented the use of the simple process algebra CCS. However, CCS turns out to be only adequate for the specification of (and reasoning on) dynamic behaviours. What was missing in this proposal was to handle data. This allows a much finer (less abstract) level of
specification, which is clearly needed in some cases. In this paper, we argue that
the process algebra LOTOS [12] and its toolbox CADP are useful respectively
to describe WSs and to reason on them. We also propose a two-level description
of WSs: an abstract one (using LOTOS) and an executable one (using WSDL
and BPEL) Following such an approach, we can develop WSs considering the
formal and verified specification as a starting point. In the other direction, we
can abstract a deployed system to a description in LOTOS. The interest of such
an abstract description is that the formality of this language and its readily ex-
sting tools enable one to validate and verify specifications through animation
and proofs of temporal properties.

To illustrate the interest of such an approach in WSs, we focus on the problem
of negotiation in which both data and dynamic aspects have to be dealt with. The
perspective of intelligent/automated WSs which would be able to automatically
perform the necessary negotiation steps to satisfy their user’s request in the
most satisfactory possible way emerged from artificial intelligence and multi-
agent systems. This problem is a typical example of services involving both
data (prices, goods, stocks, etc) and behaviours. Negotiation issues appear when
several participants (clients and providers) have to interact to reach an agreement
that is beneficial to all of them. Our goal is to show how LOTOS/CADP may
be used to ensure trustworthy and automated negotiation steps.

The organization of this paper is as follows. First, we introduce in Section
2 the different entities involved in negotiation. Section 3 presents the LOTOS
language and its toolbox CADP. They are used in Section 4 to describe nego-
tiating processes at an abstract level, and to reason on them. Section 5 gives
some guidelines to map LOTOS specifications and BPEL code in both direc-
tions. Related works are introduced in Section 6 and compared with the current
proposal. Finally, we draw up some concluding remarks in Section 7. This paper
is a shorter version of [20] in which the reader can find much more details.

2 What Does Negotiation Involve?

In this section, we introduce what is involved in negotiation cases. Specification
and implementation of such aspects are resp. described in Sections 4 and 5.

Variables. They represent entities on which processes should negotiate, e.g.
a price. Many variables may be involved in a negotiation at the same time
availability of different products, fees, maximum number of days for a delivery).

Constraints. They represent conditions to respect (called invariants as well)
while trying to reach an agreement. Such an invariant is actually a predicate
which can be evaluated replacing free variables with actual values. For a requester
who is trying to buy by auction a product, such an invariant could be that (s)he
is ready to pay €300 at most with a delivery within 10 days, or to accept a
possible late delivery if there is a price reduction of 10% at least.

Exchanged information. To reach an agreement, both participants should
send values to the other. A simple case is a price, but they can also exchange
more advanced constructs (a record of values, a constraint on a value, etc).