Normative Time-Bounded Service Logic [⋆]

E. del Val, N. Criado, M. Rebollo, and E. Argente

Grupo de Tecnología Informática - Inteligencia Artificial
Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera S/N 46022 Valencia (Spain)
{edelval,ncriado,mrebollo,eargente}@dsic.upv.es

Abstract. Traditional systems of temporal and action logics lack of an explicit representation of some desirable features in service-oriented environments, such as time and norms. In this paper a proposal for a logic which allows service-oriented environments to be formally specified is presented. This logic has been employed for formalizing normative time-bounded services in the THOMAS architecture.

Key words: Action Logics, Normative Logics, Service-Oriented Environments, State Logics.

1 Introduction

Web services and agents are autonomous entities that provide services to others, maintain their independence and modularity. In both cases they can cooperate to develop complex tasks. More concretely, the nature of agents, as intelligent and flexible entities with auto-organizative capabilities, facilitates the implementation of automatic service composition and discovering. One of the most important challenges in the computer science field is the development of open systems, which are characterized by the heterogeneity and dynamic features of both their participants and their environment. In this sense, the integration of multi-agent system (MAS) and Service Technologies has been proposed as the basis for these new and complex systems [1]. Several tools have been proposed for facilitating the development of this kind of systems. However, there is a need for logic formalisms which allow these systems to be formally specified and analysed. This paper describes a new logic framework for the representation of service-oriented environments (SOE), which provides support for the definition of quality of service parameters (e.g. temporal constraints); and deontic expressions which have been traditionally considered inside the MAS field.

On the one hand, in SOE, actions carried out by services may cause state changes. Consequently, action-based models are suitable for modelling Service-oriented systems in which actions play an important role. Action-based models

[⋆] This paper is an extended and reviewed version of a paper named "*A Logic for Normative Time-Bounded Services*" which has been presented at CAEPIA (Conference of the Spanish Association for Artificial Intelligence), in November 2009.

allow designers to specify the system properties in terms of actions or services and to reason about them. In SOE it is also desirable to reason in terms of the reachable states of the system or to consider the properties that hold in each state, i.e. the preconditions that must hold to request a service or the effects produced by the service. Therefore, it is desirable to have a model in which information can be associated to states. Service-oriented systems require a model which integrates state-based and action-based representation. On the other hand, with the proliferation of Web services as a solution to application integration, the quality of service (QoS) offered by Web services is becoming the highest priority for service providers and their partners. One of the most important properties related to QoS is service execution time. It is important to consider this information because some services could be completely useless if they are not provided on time. More specifically, time constraints are employed for representing boundaries on the service time response. Finally, SOE, as open systems, are populated by heterogeneous and autonomous agents that work together acting as service providers and consumers. Norms have been promoted inside MAS research as a mechanism for ensuring social order and dealing with unpredictability of agents. In this sense, norms define which are the expected behaviours in terms of deontic statements such as obligations, permissions or prohibitions.

A new logic framework, known as Normative Time-bounded Service Logic (NTBSL), which covers the above mentioned needs of SOE, is presented in this paper. It is suitable for the formal specification of SOE in which both agents and Web services cooperate for developing complex systems. The proposed logic allows an explicit representation of services as actions and also the specification of temporal boundary constraints and normative expressions in the states of the system. This paper is organized as follows: a review of related works is contained in Section 2. In Section 3, the background for NTBSL is presented. Section 4 presents syntax and semantics of NTBSL. In Section 5, an architecture based on the previous formalization is described, detailing the management of services and norms in the architecture through an example. Finally, in Section 6 conclusions and future work are described.

2 Related Works

Works on norms inside the MAS field normally have a theoretical point of view, describing logical relationships among obligations, prohibitions and permissions by means of deontic logics [2, 3]. Regarding works belonging to the community of semantic web, several proposals have been done in order to define policies or regulations over the system performance. A well known example of these works is the KAoS policy services framework [4]. It is a service-oriented architecture which provides functionalities for managing and controlling policies. This proposal allows deontic constraints to be explicitly represented as terms belonging to an OWL ontology. However, it lacks a logic formalism which includes both normative and temporal QoS constraints; that would allow SOE to be specified and analysed.

This logic formalism is needed to provide specifications interpreted over models, which describe the system properties. With this aim, different types of logics have been proposed. In this sense, modal and temporal logics deal with notions such as necessity, possibility, eventuality, etc. For instance, CTL[5] allows the system properties to be expressed in terms of their states. Regarding mixed systems, which are described in terms of both actions and states, it is more natural to use an action-based temporal logic to express their properties. In order to model mixed systems, several logics have been proposed such as ARCTL[6]. ARCTL is interpreted over a Mixed Transition System (MTS)[7] model and can specify a system in terms of its actions and states. It has the same temporal operators as CTL, except that they can be restricted to paths whose actions satisfy a given action formula α. This fact makes ARCTL a suitable logic for reasoning in terms of actions, since it is a logic which can be interpreted over Labelled Transition Systems (LTS)[8], and also for reasoning and checking properties over states since it is based on the modal logic CTL and can be interpreted over Kripke Structures (KS)[9]. Despite the fact that the ARCTL formalism gives support for reasoning about SOE in terms of both actions and states, it lacks of an explicit representation of some desirable features in SOE, such as temporal and deontic constraints.

Considering the requirements of SOE, NTBSL extends previous works on deontic logics, developed inside the MAS community, and MTS proposals. In this sense, NTBSL is proposed as a logic formalism for supporting the specification of open systems based on the integration between service and MAS technologies.

3 Background

NTBSL extends ARCTL by including time and norms and it is interpreted over a MTS. This section describes its logic background and the basis of MTS.

3.1 Logical Basis for NTBSL

On the one hand, in order to provide an explicit representation of time and probability, NTBSL uses the same mechanism that PCTL[10] for representing explicit time bounds and probabilities. The explicit time representation in NTBSL allows reasoning about time properties in real systems such as *soft deadlines. Soft deadlines* are useful in systems in which having a bounded response time is important, but the failure to meet the response time does not result in a disaster for the system. There is a set of logics based on CTL which deals with quantitative time such as PCTL[10], RTCTL[11], vRTCTL or PRTCTL[12]. PCTL tags temporal operators with time bounds, as in RTCTL. PCTL also provides methods for reasoning about time such as *"after a request for a service, the service will give an answer within 2 seconds"*. In addition, for enabling reasoning about *soft deadlines*, state expressions are tagged with probabilities. Therefore, expressible properties are: "with at least 50 percent probability p will hold within 20 time units" $[F^{\leq 20}p]_{\geq 0.5}$ or "with at least 99 percent probability q will hold continuously for 20 time units" $[F^{\leq 20}p]_{\geq 0.99}$.

On the other hand, NTBSL allows expressing norms which control service performance. Our proposal of extending temporal logics for representing norms is based on the Normative Temporal Logic (NTL)[13], a formalism intended for reasoning about the temporal properties of normative systems. NTBSL introduces deontic operators in order to give support to the definition of norms.

As stated before, NTBSL is interpreted over a mixed transition models. Next, an overview of mixed transition systems models is provided.

3.2 Mixed Transition Systems

Mixed Transition Systems (MTS) models integrate state-based models (Kripke structures, KS) and action-based models (Labelled Transition Systems, or LTS) into a common super-structure. Given two sets of propositional atoms P_S and P_A, over states and actions respectively; a MTS over P_S and P_A is a structure $M = \langle S,\, S_0,\, A,\, T,\, V_S,\, V_A\rangle$, where:

- S is a non-empty set of states;
- $S_0 \subseteq S$ is the set of possible initial states;
- A is a non-empty set of actions;
- $T \subseteq S \times A \times S$ is the transition relation;
- $V_S : S \rightarrow 2^{P_S}$ is the interpretation function on states;
- $V_A : A \rightarrow 2^{P_A}$ is the interpretation function on actions.

MTS combines actions over transitions from LTS and propositional atoms over states from KS, and adds propositional atoms over actions that allow for a generalized and more uniform presentation of logic formulas over MTS models. An MTS can be projected to a KS sub-structure $\langle S,\, S_0,\, R,\, V_S\rangle$, where $R = \{(s,\, s'\ ) \mid (s,\, a,\, s') \in T\ \}$, or an LTS sub-structure $\langle S,\, S_0,\, A,\, T\ \rangle$, and thus both state-based and action-based logics can be interpreted over an MTS.

A path π of M is a finite or infinite sequence of connected transition steps $(s_{i-1}\ ,\ a_i\ ,\ s_i\ ) \in T$, denoted as $s_0 \overset{a_1}{\rightarrow} s_1 \overset{a_2}{\rightarrow} s_2\ ...\ \overset{a_n}{\rightarrow} s_n$. In particular a zero length path consists of a single state. Let $T*$ be the set of finite paths of M and T^w the set of infinite paths. Given a path $\pi = s_0 \overset{a_1}{\rightarrow} s_1 \overset{a_2}{\rightarrow} s_2\ ...\ \overset{a_n}{\rightarrow} s_n$, is defined:

- $|\pi|$ = n (resp. φ), the length of a path;
- $\pi(i) = s_i$, the i-th state of π $(0{\leq}i{\leq}|\pi|)$;
- $\pi(\bullet i) = a_i$, the i-th action of π $(1{\leq}i{\leq}|\pi|)$;

A full-path is a path that is either infinite or ends in a terminal state; being a terminal state those one with none successor. $\Pi(M)$ (or just Π) is defined as the set of full-paths of M, and $\Pi(M,\, s)$ (or $\Pi(s)$) as the set of full-paths from state s.

$$\Pi(M){:=}\ T^{\omega}{\cup}\{\pi{\in}T{*} \mid (|\pi| = n \wedge \pi(n) \nrightarrow)\}$$
$$\Pi(s){:=}\{\pi \in \Pi(M) \mid \pi(0) = s\}$$

Considering M as a model, a *normative system* η is a set of transitions which are labelled as forbidden ($\eta \subseteq T$) [13]. More concretely, a normative system defines, for every possible system transition, whether or not that transition is considered to be legal or not. $T \backslash \eta$ is the original model without the forbidden transitions. It is a total relation. This requirement is a reasonable constraint: it prevents normative systems which lead to states with no successor. The *implementation of a normative system* η over a model M is a new model $M_\eta = \langle S, S_0, A, T', V_S, V_A \rangle$ where $T' = T \backslash \eta$.

Once the logic background and the MTS models have been described, the proposed logic framawork, named Normative Time-bounded Service Logic, is defined in the next section.

4 Normative Time-bounded Service Logic

The Normative Time-Bounded Service Logic, or NTBSL for short, takes ARCTL as a reference and extends it by supporting the definition of temporal and normative constraints on the performance of services. Following, syntax and semantics of NTBSL are detailed.

4.1 NTBSL Syntax

The syntax of NTBSL is defined by the formulas generated by the following grammar, where $q \in P_S$ and ϕ range over state formulas; γ is a path formula; $b \in P_A$ and α are action formulas. State formulas (S1-S9) represent properties of states. In S7-S9 productions the normative operators O, P and Fo refer to *obligation*, *permission* and *prohibition*, respectively. Expressions contained in A1-A4 are the usual productions for the definition of action expressions. Path formulas (P1-P3) represent properties of paths. In path productions, operators U, F and G refer to traditional temporal operators *Until*, *Finally* and *Globally*, respectively. The parameters t and p represent time constraints and the probability of the path formula, respectively.

$$
\begin{aligned}
(S1-S9)\ \phi ::= {} & \mathsf{true} \mid q \mid \neg\phi \mid \phi \wedge \phi \mid \mathsf{A}_\alpha\,[\gamma]_{\geq p} \mid \mathsf{E}_\alpha\,[\gamma]_{\geq p} \mid \mathsf{O}^\eta_\alpha\,[\gamma]_{\geq p} \mid \mathsf{P}^\eta_\alpha\,[\gamma]_{\geq p} \mid \\
& \mathsf{Fo}^\eta_\alpha\,[\gamma]_{\geq p} \\
(A1-A4)\ \alpha ::= {} & \mathsf{true} \mid b \mid \neg\alpha \mid \alpha \wedge \alpha \\
(P1-P3)\ \gamma ::= {} & \phi\,\mathsf{U}^{\leq t}\,\phi \mid \mathsf{F}^{\leq t}\,\phi \mid \mathsf{G}^{\leq t}\,\phi
\end{aligned}
$$

Derived forms such as $\mathsf{E}_\alpha\mathsf{F}$ are defined similarly as in CTL. Intuitively, given a NTBSL formula $\mathsf{E}_\alpha[\gamma]_{\geq p}$, the path formula γ is evaluated over full α-prefixes of full-paths of the model. To formalize that, the α-restriction of a MTS is defined as the structure $M|_\alpha = \langle S, S_0, A, T|_\alpha, V_S, V_A \rangle$ where $T|_\alpha = \{(s, a, s') \in T : a \models \alpha\}$. A_α and E_α are interpreted over the full-paths of $M|_\alpha$, and the path formulas are defined as in standard CTL.

4.2 NTBSL Semantics

The semantic relation over states ($(M,\, s) \models \phi$) or paths ($(M,\, \pi) \models \gamma$) are defined as follows :

$(S1)\ (M,s) \models \mathsf{true}$

$(S2)\ (M,s) \models q$ iff $q \in V_S(s)$

$(S3)\ (M,s) \models \neg\phi$ iff $(M,s) \not\models \phi$

$(S4)\ (M,s) \models \phi \wedge \phi'$ iff $(M,s) \models \phi \wedge (M,s) \models \phi'$

$(S5)\ (M,s) \models \mathsf{A}_\alpha[\gamma]_{\geq p}$ iff $\forall\ \pi \in \Pi(M|_\alpha, s) \cdot (M,\pi) \models \gamma\ ,\ \mu_m \geq p$

$(S6)\ (M,s) \models \mathsf{E}_\alpha[\gamma]_{\geq p}$ iff $\exists\ \pi \in \Pi(M|_\alpha, s) \cdot (M,\pi) \models \gamma\ ,\ \mu_m \geq p$

$(S7)\ (M,s) \models \mathsf{P}^\eta_\alpha[\gamma]_{\geq p}$ iff $(M_\eta, s) \models \mathsf{E}_\alpha[\gamma]_{\geq p}$

$(S8)\ (M,s) \models \mathsf{O}^\eta_\alpha[\gamma]_{\geq p}$ iff $(M_\eta, s) \models \mathsf{A}_\alpha[\gamma]_{\geq p}$

$(S9)\ (M,s) \models \mathsf{Fo}^\eta_\alpha[\gamma]_{\geq p}$ iff $(M_\eta, s) \models \mathsf{A}_\alpha[\neg\gamma]_{\geq p}$

$(P1)\ (M,\pi) \models \phi\, \mathsf{U}^{\leq t}\, \phi'$ iff $\exists i \in [0,t] \cdot (M, \pi(i)) \models \phi'\ \wedge$
$\forall k \in [0, i-1] \cdot (M, \pi(k)) \models \phi$

$(P2)\ (M,\pi) \models F^{\leq t}\phi$ iff $\exists i \in [0,t] \cdot (M, \pi(i)) \models \phi$

$(P3)\ (M,\pi) \models G^{\leq t}\phi$ iff $\forall i \in [0,t] \cdot (M, \pi(i)) \models \phi$

Semantics of expressions contained in S1-S4 is defined as usual. In S5, $A_\alpha[\gamma]_{\geq p}$ expresses that, for all the prefixes of α, γ holds with a probability equal or greater than p. The expression in S6, $E_\alpha[\gamma]_{\geq p}$, means that it exists at least one prefix of α, for which γ holds with a probability equal or greater than p. The function $\mu_m : 2^S \rightarrow [0,1]$ is a finitely additive probability measure on 2^S; for its definition see [10]. Semantics of the deontic expressions (S7-S9), which express obligations, permissions and prohibitions, have been defined as in NTL. For example a permission formula, $\mathsf{P}^\eta_\alpha[\gamma]_{\geq p}$ (S7), means that in the *normative implemented model* (M_η) there is an α-path which satisfies $[\gamma]_{\geq p}$. Similarly, an obligation formula $\mathsf{O}^\eta_\alpha[\gamma]_{\geq p}$ is true when all the α-paths over the implementation of the normative system (M_η) satisfy the formula $[\gamma]_{\geq p}$. Path formulas are modified to include temporal constraints over temporal operators U, F and G. More specifically, P1 expression means that the formula ϕ, in the state $\pi(k)$, will be true from now until instant t when the formula ϕ' in state $\pi(i)$ becomes true. In P2 the expression means that the formula ϕ holds within t time units with a probability of at least p. The expression in P3 means that the formula ϕ holds continuously for t time units.

As claimed above, semantic of deontic operators (O, P and Fo) are defined by means of path quantifiers operators (A and E). However, there is a main a difference among deontic and path formulae. The former are interpreted over the normative implemented model (M_η), which represents the ideal world. On the contrary, path formulas are interpreted over the full model (M) which represents what actually holds.

NTBSL, presented in this section allows to specify service-oriented approaches. This logic is interpreted over MTS thus it allows specification over states and transitions, as well as the specification of time-bounded services as actions with deadlines and normative expressions in states, which can be viewed as the service preconditions and effects. In the following section an architecture based on this formalization is presented.

5 Architecture For Normative Time-bounded Services

Based on the previous logic, in this section the THOMAS architecture which supports the concept of normative time-bounded service is described. THOMAS [14] is an open architecture that employs a service-based approach as the basic building blocks for creating a suitable platform for intelligent agents grouped in Virtual Organizations (VOs). This architecture consists of three components (see Figure 1):

- *Platform Kernel* (PK), that deals with basic agent management services and it can be provided by any FIPA-compliant platform. Its functionality is related with the agent life-cycle and the network communication layer.
- *Service Facilitator* (SF), which is a service manager that registers services provided by external entities. These services are based on the concept of normative time-bounded service. Furthermore, the SF facilitates service discovering for potential clients. It can be considered as a yellow pages server.
- *Organization Management System* (OMS), which is responsible for the management of virtual organizations, taking control of their underlying structure, the roles played by the agents inside the organization and the norms that rule the system behaviour.

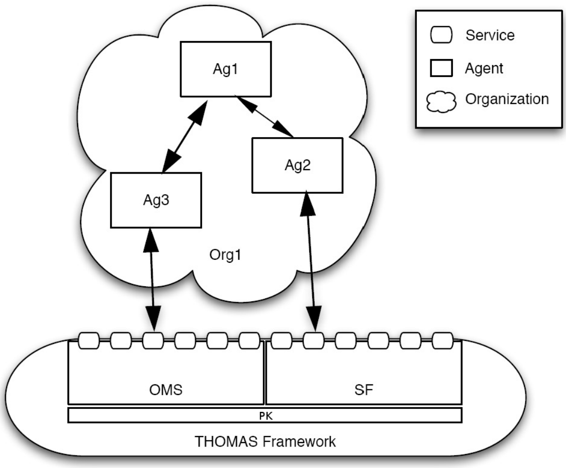

Fig. 1. Thomas service-based architecture

Following the application of the NTBSL logic in both the service and norm management provided by the THOMAS architecture is detailed.

5.1 Service Management

The service management is carried out by the SF component. This component is capable of dealing with services in a more elaborated way, following *Service Oriented Architectures* guidelines. Thus, the SF copes with information related to norms, service execution time, service roles, goals as well as service composition, based not only on Inputs and Outputs but also on Preconditions and Effects.

The organizations and agents in THOMAS offer their functionality through services. A service in THOMAS is considered as a normative-time-bounded service, i.e. it is a service whose estimated response time is known, it uses probability to model *soft deadlines* and norms can appear in its preconditions and effects.

NTBSL logic allows us to represent these features, to reason about service behaviour (actions) and also to reason about the preconditions and effects (states). A normative-time-bounded service in NTBSL could be specified as an action α in a transition $(\pi_k,\alpha,\pi_i) \in T$ of a path π and is characterized by: a set of preconditions and effects in states π_k and π_i respectively, a *soft deadline* t and the probability of that deadline p.

$$\begin{array}{c} A_\alpha[\phi \ \mathsf{U}^{\leq t} \ \phi']_{\geq p} \\ (M, \pi(k)) \models \phi \\ (M, \pi(i)) \models \phi' \end{array}$$

If in state π_k service inputs and preconditions are satisfied (ϕ) and if service α is executed, then effects (ϕ') hold in the state π_i in less than t time units with a probability of at least p. With this model it is possible to express properties such us *"with probability at least 0.85 once the preconditions are fulfilled, the desired effects will be satisfied through the execution of a* $serv_1$ *in at most 23 time units"*:

$$A_{serv_1}[preconditions \ \mathsf{U}^{\leq 23} \ effects]_{\geq 0.85}$$

The above formula claims that the effects of service $serv_1$ will be obtained at most 23 time units with probability at least 0.85 in all situations. Following, an example of the usage of the NTBSL for representing both norms and normative-time-bounded services in THOMAS is detailed.

5.2 Normative Management

The THOMAS architecture allows defining norms that prescribe agent rights and duties in terms of who can provide a service, when and under which circumstances. In addition, norms can be viewed as a coordination skill for organizing MAS, since they specify the desired behaviour of the society members [15]. More specifically, norms define role functionality in terms of services that can be requested/provided, service requesting order, service conditions and interaction protocols that should be followed. For a detailed description of these norms and the normative language employed in the THOMAS architecture see [16]. In this section, a simplified version of this language which allows to express temporal constraints on the provision of services is proposed.

Normative Language Syntax. Next, a simplified description of the developed normative language is shown.

$$
\begin{aligned}
< \mathit{norm} > &::= < \mathit{normative\_system} >< \mathit{deontic\_control} > \\
&\quad \mathsf{PROVIDE} < \mathit{service} > \\
&\quad [< \mathit{temporal\_constraint} >]\ [< \mathit{probability} >] \\
< \mathit{normative\_system} > &::= \mathit{id} \\
< \mathit{deontic\_control} > &::= \mathsf{FORBIDS} \mid \mathsf{PERMITS} \mid \mathsf{OBLIGES} \\
< \mathit{service} > &::= ([\mathit{precondition},]\ \mathit{service\_id}\ [,\mathit{postcondition}]) \\
< \mathit{temporal\_constraint} > &::= \mathsf{BEFORE}\ \mathit{deadline} \\
< \mathit{probability} > &::= \mathsf{PROBABILITY}\ \mathit{probability}
\end{aligned}
$$

In concrete a norm belonging to a normative system defines a deontic control concerning the performance ($(\mathit{precondition}, \mathit{service\_id}, \mathit{postcondition})$) of a specific service which must be provided within a period of time ($< \mathit{temporal\_constraint} >$) with a concrete probability.

Normative Language Semantics. The semantics of this normative language can be defined by means of the NTBSL. Thus, a prohibition norm is specified as follows:

$$
\begin{gathered}
\eta\ \mathsf{FORBIDS\ PROVIDE}\ (\mathit{precondition}, \mathit{service}, \mathit{postcondition}) \\
\mathsf{BEFORE}\ \mathit{deadline}\ \mathsf{PROBABILITY}\ \mathit{probability} \\
\updownarrow \\
\mathsf{Fo}^{\eta}_{\mathit{service}}[\mathit{precondition}\ \mathsf{U}^{\leq \mathit{deadline}}\ \mathit{postcondition}]_{\geq \mathit{probability}}
\end{gathered}
$$

Similarly, semantics for obligation and permission norms are defined by means of the deontic (O and P) operators belonging to NTBSL, as follows:

$$
\begin{gathered}
\eta\ \mathsf{OBLIGES\ PROVIDE}\ (\mathit{precondition}, \mathit{service}, \mathit{postcondition}) \\
\mathsf{BEFORE}\ \mathit{deadline}\ \mathsf{PROBABILITY}\ \mathit{probability} \\
\updownarrow \\
\mathsf{O}^{\eta}_{\mathit{service}}[\mathit{precondition}\ \mathsf{U}^{\leq \mathit{deadline}}\ \mathit{postcondition}]_{\geq \mathit{probability}}
\end{gathered}
$$

$$
\begin{gathered}
\eta\ \mathsf{PERMITS\ PROVIDE}\ (\mathit{precondition}, \mathit{service}, \mathit{postcondition}) \\
\mathsf{BEFORE}\ \mathit{deadline}\ \mathsf{PROBABILITY}\ \mathit{probability} \\
\updownarrow \\
\mathsf{P}^{\eta}_{\mathit{service}}[\mathit{precondition}\ \mathsf{U}^{\leq \mathit{deadline}}\ \mathit{postcondition}]_{\geq \mathit{probability}}
\end{gathered}
$$

5.3 Example

Considering the definition of a normative-time-bounded service, the service description, apart from the usual information related to Inputs, Outputs, Preconditions and Effects (IOPEs), should be modified in order to contain an explicit representation of time, probability and norms. These parameters facilitate the SF to discover and composite tasks between a set of registered services. The SF can know not only whether a service provides the required functionality but also whether the client is allowed to request the service, considering the active norms, or if the service can be provided before a client's deadline.

Individual agents or organizations, in order to offer this information should register their services in the SF. The service description is an OWL-S service description, extended with non-functional parameters. This description introduces two non-functional parameters which represent service execution time and probability. The value of the service execution time is represented by an average execution time of the service. This execution time is obtained by measuring the cost of the service off-line. Besides this, preconditions and effects can contain normative expressions, which allow expressing the conditions that must or may hold before/after providing a service.

Norm Example. Let us suppose that in a critical situation a service must be provided within a restricted deadline, whereas in a normal situation the temporal duration is less restrictive. This kind of conditional constraints is expressed by means of norms as follows:

$$\begin{gathered}\eta_{critic}\ \mathsf{OBLIGES\ PROVIDE}\ (unknownUser, login, logged(user, password))\\ \mathsf{BEFORE}\ 10\ \mathsf{PROBABILITY}\ 1\\ \updownarrow\\ O^{\eta_{critic}}_{login}[unknownUser\ \mathsf{U}^{\leq 10}\ logged(user, password)]_{\geq 1}\end{gathered}$$

More specifically, this norm claims that according to the normative system which corresponds to a critical situation ($\eta_{critical}$) the login service (*login*) must be provided within 10 temp units.

Service Description Example. As argued before, the SF needs a more complete service description to select the most suitable service or set of services according to the current state (i.e. the normative system). This extra information has been expressed using non-functional parameters. In addition normative prescriptions which describe preconditions and postconditions of services have been described by means of the proposed normative language.

Figure 2 contains the non-functional parameters of a *login* service. As specified in the service description, the service provider carries out the login service within 25 temporal units in the 85% of the cases. Figure 3 illustrates an example of postcondition for this login service. More specifically, this postcondition contains a norm expressed in NTBSL. It states that after carrying out the login service the client will be permited to provide its own *Information* service inside

```
<process:AtomicProcess rdf:ID="login">
...
<time:hasLocal>
  <time:time-Expression rdf:ID="S-Time">
  <expr:expressionBody rdf:datatype="http://...#long">
  25
  </expr:expressionBody>
  </time:time-Expression>
</time:hasLocal>
<probability:hasLocal>
  <probability:probability-Expression rdf:ID="S-Probability">
  <expr:expressionBody rdf:datatype="http://...#float">
  0.85
  </expr:expressionBody>
  </probability:probability-Expression>
</probability:hasLocal>
...
```

Fig. 2. Non-functional parameters in an OWL-S real-time service description

THOMAS. The symbol $\eta_{\emptyset}$ corresponds to the normative system in which none transition is considered as forbidden (i.e. $\eta_{\emptyset} = \emptyset$).

Therefore, the SF component of the THOMAS architecture is able to select the most appropriated service considering both the temporal and state conditions which must be satisfied according to the normative system which is active in a certain moment. In this example, the *login* service (whose description is contained in Figures 2 and 3) is not legal according to the normative system corresponding to a critical situation (η_{critic}); since in event of a critic situation the temporal deadline is more restrictive (the service must be provided in 10 temporal units).

```
<process:AtomicProcess rdf:ID="login">
...
<process:hasPostcondition>
 <ntbsl:NTBSL-Condition rdf:ID="NTBSL-Precondition">
  <expr:expressionBody rdf:datatype="http://...#string">
   $\eta_{\emptyset}$ PERMITS PROVIDE (Information)
  </expr:expressionBody>
 </ntbsl:NTBSL-Condition>
</process:hasPostcondition>
...
</process:AtomicProcess>
```

Fig. 3. Postcondition of an OWL-S real-time service description

6 Conclusions

In SOE it is desirable to reason about the properties that hold at each reachable state and to reason in terms of the actions that can be performed by services. Another desirable feature of formal models for SOE is the possibility of reasoning about temporal and normative constraints. In this paper, we propose an extension of previous works on the integration of both state-based and action-based models by giving support to the explicit representation of time, probability and the deontic operators. This proposed logic has been applied in the THOMAS architecture. All of the functionalities in THOMAS are described and provided as normative time-bounded services. As future work we plan going on the development of service-oriented systems by considering the problem of reasoning about the system properties expressed by means of NTBSL.

Acknowledgements

This work is supported by TIN2005-03395 and TIN2006-14630-C03-01 projects of the Spanish government, GVPRE/2008/070 project, FEDER funds and CONSOLIDER-INGENIO 2010 under grant CSD2007-00022, FPU grant AP-2007-01256 awarded to N.Criado, FPU grant AP-2008-00601 awarded to E. del Val.

References

1. M. Luck, P. McBurney. Computing as Interaction: Agent and Agreement Technologies. In Proceedings of the IEEE SMC Conference on Distributed Human-Machine Systems, 1-6, 2008. (Invited Paper)
2. F. Dignum. Autonomous agents with norms. In *Artif. Intell. Law*, 7(1):6979, 1999.
3. M.J. Sergot. A computational theory of normative positions. In *ACM Trans. Comput.Log.*, 2(4): 581622, 2001.
4. A. Uszok, J. Bradshaw, R. Jeffers, N. Suri, P. Hayes, M. Breedy, L. Bunch, M. Johnson, S. Kulkarni, J. Lott. KAoS Policy and Domain Services: Toward a Description-Logic Approach to Policy Representation, Deconfliction, and Enforcement,". In *POLICY*, pp. 93, 2003.
5. E. A. Emerson. Temporal and modal logic. In *Handbook of theoretical Computer Science* (Vol. B): Formal Models and Semantics, J. van Leeuwen, Ed. MIT Press, Cambridge, MA, 995-1072, 1990.
6. C. Pecheur and F. Raimondi. Symbolic model checking of logics with actions. *Model Checking and Artifical Intelligence: 4th Workshop, MoChArt IV, Riva del garda, Revised Selected and Invited Papers*, pages 113–128, 2006.
7. O. Wei, A. Gurfinkel, M. Chechik. Mixed Transition Systems Revisiteds. In *VMCAI*, LNCS 5403:349-365, 2009.
8. J.-P. Katoen. Labelled Transition Systems. In *Model-Based Testing of Reactive Systems* LCNS 3472: 615-616, 2004.
9. S. Kripke. Semantical considerations on modal logic. In *Philosophica Fennica* 16: 83-94, 1963.
10. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:102–111, 1994.

11. E. A. Emerson, A. P. Sistla and J. Srinivasan. Quantitative Temporal Reasoning. *Proc. of the Workshop on Automatic Verification Methods for Finite State Systems*, 136–145, 1989.
12. E. A. Emerson and R. J. Trefler. Parametric quantitative temporal reasoning. *Logic in Computer Science, Symposium on*, 0:336, 1999.
13. T. Agotnes, W. V. der Hoek, J. Rodriguez-Aguilar, C. Sierra, and M. Wooldridge. On the logic of normative systems. *IJCA*, AAAI Press, pages 1175–1180, 2007.
14. C. Carrascosa, A. Giret, V. Julian, M. Rebollo, E. Argente, and V. Botti. Service Oriented MAS: An open Architecture. In *AAMAS*, pp. 1-2, 2009.
15. G. Boella, L. van der Torre, H. Verhagen. Introduction to the special issue on normative multiagent systems. *Auton. Agents Multi-Agent Syst.*, 17:1–10, 2008.
16. E. Argente, N. Criado, V. Julian, V. Botti. Designing Norms in Virtual Organizations. In *CCIA*, 184:pages 16–23. IOS Press, 2008.