Process Modelling: The Deontic Way

Vineet Padmanabhan[1], Guido Governatori[1], Shazia Sadiq[1], Robert Colomb[1] & Antonino Rotolo[2]

[1] School of Information Technology & Electrical Engineering
The University of Queensland, Queensland, Australia
Email:[vnair,guido,shazia,colomb]@itee.uq.edu.au
[2] CIRSFID, University of Bologna, Bologna, Italy
Email: rotolo@cirsfid.unibo.it

Abstract

Current enterprise systems rely heavily on the modelling and enactment of business processes. One of the key criteria for a business process is to represent not just the behaviours of the participants but also how the contractual relationships among them evolve over the course of an interaction. In this paper we provide a framework in which one can define policies/ business rules using deontic assignments to represent the contractual relationships. To achieve this end we use a combination of deontic/normative concepts like *proclamation*, *directed obligation* and *direct action* to account for a deontic theory of commitment which in turn can be used to model business processes in their organisational settings. In this way we view a business process as a *social interaction process* for the purpose of doing business. Further, we show how to extend the i^* framework, a well known organisational modelling technique, so as to accommodate our notion of deontic dependency.

Keywords: Business, Enterprise and Process modelling.

1 Introduction

One of the issues which was heavily debated during the panel discussions of AAMAS-04 (Singh 2004) was with regard to considering business process modelling/management as a killer application for agents. It was decided that in-order to make progress in this direction it was inevitable to characterise problems within business process modelling/management where agents can apply. In this paper we make such a move whereby we consider an agent to represent a real world business partner with its own local business rules and configurations. The collaboration between the different partners is made possible through *normative co-ordination*, i.e., the idea that agents can achieve flexible co-ordination by conferring normative positions like duties, permissions and powers to other agents. In this way we can view the partners involved in a business scenario as multiple agents who might collaborate by creating *commitments* using normative concepts such as obligation, proclamation and so on but at the same time retain their autonomy. This in a way is similar to how a Multi-Agent System (MAS) is defined using abstractions like team and commitments from the perspective of organisations and societies (Conte & Dellarocas 2001, Pitt 2005).

The definition of a business process we adopt in this paper is more general in the sense that we consider a business process as a special kind of *social interaction process*. A ***social interaction process*** *is a temporally ordered, coherent set of events and actions, involving one or more communication acts that may create commitments, perceieved and performed by agents, and following a set of rules, or protocol, that is governed by norms, and that specifies the type of the interaction process* (Wagner 2003).

Based on the above definition of a business process, we develop a logical framework based on multi-modal logic to capture the normative positions among agents in an organisational setting. We do not take into account any temporal considerations in this work as it remains part of future work. Also commitment is not taken as a primitive but is rather defined in terms of directed obligations, i.e., an agent's obligation towards another agent. For example, the obligation to pay for delivered goods is directed, so to say from the buyer to the seller in a business situation like trade where different agents like *buyer, seller, transport company, customs offices* etc. exist. The reasons for not adopting commitments as a primitive is that (1) we want to show that deontic logic can be used to capture an agents *obligation* to another and (2) thereby show that the argument made in (Singh, Chopra, Desai & Mallya 2004) which states that *commitments fare better than traditional deontic logic because deontic logic disregards an agents obligation to another agent* is not true. Two other important concepts we use in our framework are that of *proclamation* and *direct action*. Proclamation is seen as a particular type of speech-act (communication act) that helps the agents to create normative positions involving other agents. It is similar to the role speech acts play in *language-action* approaches to work-flow management as given in (Goldkuhl 1996). Proclamation is used to cover all those speech acts by which an agent makes a statement expressing a certain proposition with the aim of making the proposition true. In other words, different types of speech acts like *directives* (permitting, requesting, forbidding etc.), *Commissives* (agreeing, offering, promising etc.), *Constatives* (announcing, disagreeing, informing etc.) etc. are considered as instances of just one speech act like proclamation. The advantage of this approach is that one need to worry only about the specific semantics corresponding to the *proclamation* operator rather than for each type of speech act. On the other hand direct action is concerned with relationships between agents and states of affairs they want to realise/bring about. We discuss more about these in the coming sections.

Let us put together the different concepts explained above with the help of a an example. Consider the work-flow in Figure 1. depicting a simple scenario of managing after sales service.

This process is assumed to execute in a technology which represents a shared space between various business partners and stake-holders of the overall business process. The 4 actors (roles) within the scenario are that of *Customer*, *Retailer*, *Manufacturer* and *Technician*. In such a set up it is natural to think of commitments as being made from one actor to another to achieve some goal and these commitments could be in the form of obligations. For ex-

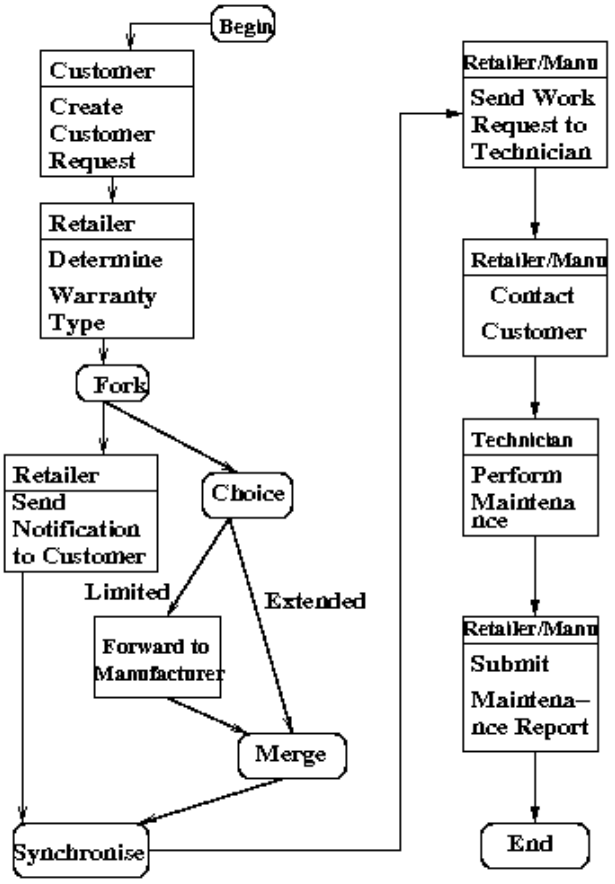

Figure 1: After sales service work-flow

ample, *Send work request to technician* and *Contact Customer* are two activities which may be allocated to either the retailer or the manufacturer depending on the particular instance. Sending request to technician may involve an internal activity of first finding the most appropriate technician. For instance, the retailer/manufacturer is normally *obliged* to issue several RFQs (Request For Quote) to several technicians and choose from among them. The reason is that the retailer/manufacturer wants to get the best quote. The technicians are *obliged* to respond to an RFQ because it represents a potential business that he/she needs. Hence we can say that a particular technician t undertakes an obligation towards the retailer/manufacturer to perform job j while the retailer/manufacturer undertakes towards the technician the obligation of paying the price p. Such obligations are called *directed* obligations and we explain more about their logical properties in section 2. In this way we can define commitments in terms of directed obligations from one party to another. In general, if we have two actors (roles) representing a *debtor* and the other a *creditor* in a particular exchange relationship it is natural to think of a commitment as a directed obligation from the debtor to the creditor regarding a particular condition that in effect the debtor promises to *bring about*. It is also the case that the different actors could have reciprocal obligations whereby they can make effective contractual relationships/joint commitments between them. We show how a combination of notions related to *obligation, proclamation* and *action* accounts for a deontic theory of commitment which in turn can be used to model business processes in their organisational settings. Not only does such a model enable to describe and analyse commitments that exist between the actors of a problem domain but also helps to achieve coherent behaviour in interaction processes. We achieve this by defining various operations that can be performed to create and manipulate commitments that would help the different actors involved to create various normative relations needed to co-ordinate their behaviours for the overall success of a business process. In this way, similar to (Taveter & Wagner 2001*a*), we view a business process as a social interaction process for the purpose of doing business.

Our *deontic approach* to process modelling/management is based on works like (Barbuceanu, Gray & Mankovski 1999, Taveter & Wagner 2001*a*, Yu & Mylopoulos 1993, Yu & Mylopoulos 1994) where it is often argued that organisations are made up of social *actors* who have goals and interests which they pursue through a network of relationships with other actors[1]. Hence a *richer model of business process should include not only how work products (entities) progress from process step to process step (activities) but also how the actors performing these steps relate to each other* intentionally*, i.e., in terms of concepts such as goal, belief, commitment etc.*. Such process models conveys a deeper understanding of a business process by focus sing on the intentional dependencies among actors which is extremely important for modelling business processes between enterprises that consists of the steps of analysis and design. Following this paradigm, in this work, we develop a formal representation based on a deontic approach to capture the normative dependencies between the different actors in a business process scenario. The formal representation allows us to achieve coherent behaviour in the interaction process where the rules of engagement are dynamically and frequently changing.

The paper is structured as follows. In section 2 we outline the various ingredients of our logical framework needed to represent the different aspects involved in a social interaction process (business process). In section 3 we propose our theory of commitment based on a combination of proclamation, obligation and direct action. The next section (section 4) demonstrates an example scenario in which we show our working model. Section 5 shows some general rules of engagement between different actors in an organisational setting. Section 6 makes a comparison of our model with that of i^* and extends the later to accomodate normative dependency. Section 7 talks about related as well as future work.

2 Institutional Agency

As mentioned in the previous section business processes exist in social organisational settings wherein interaction between agents takes place in a social context. Hence normative concepts are essential for understanding and controlling coherent interaction between agents and other systems. For this paper we take as background the well-known Kanger-Lindahl-Pörn (Kanger 1972, Lindahl 1977, Pörn 1977) logical theory to account for agency and organised interaction (see (Elgesem 1997)). Our starting point is to take advantage of some recent contributions (Santos, Jones & Carmo 1997, Jones & Sergot 1996, Pitt 2005), which have enriched this framework with some substantial refinements. As we have alluded to, the notion of agency is described in a multi-modal logical setting. Despite some well-known limitations (see (Elgesem 1997, Royakkers 2000)), such an approach is very general since actions are simply taken to be relationships between agents and states of affairs, and very flexible since it allows for the easy combination of actions and concepts like powers, obligations, beliefs, etc. It also permits to provide a simple conceptual analysis of the structure of organisations of agents. As recently pointed out regarding the design of computerised multi-agent systems, such a multi-modal logic "[is] a means of supplying an intermediate level of description, falling somewhere between [...] ordinary-language account of what a system [...] is supposed to be able to do and [...] the level of implementation" (Pitt 2005).

[1] We use the term agent/actor interchangeably.

2.1 The Logical Framework

We first provide the basic ingredients that make up a theory of institutional agency and later show how a notion of commitments can be captured by combining these basic ingredients. We start with the idea of personal and direct action to realise a state of affairs, formalised by the modal operator E: E_xA means that the agent x brings it about that A. For example, suppose that A represents a situation in Figure(1) where *a particular technician, t_1, can do the maintenance job on a specific date*. Then it could be that the retailer r brings it about that A by sending a request to the technician t_1. Different axiomatisations have been provided for E but almost all include the following schemas.

$$E_xA \rightarrow A \tag{1}$$

(1) is recognised as valid by almost all theories of agency. It is nothing but the usual axiom T of modal logic, and it expresses the successfulness of actions that is behind the common reading of "bring about" concept. We reformulate this axiom as $E_XA \rightarrow A$ to represent a set of agents X. $E_{\{x\}} = E_x$ when the set of agents is a singleton. E_xA can have the form $E_x sendGoods(p)$, (where A is an action predicate denoting a specific action), meaning agent x sends the goods p. Here agent x executes by itself the action A. Most of the examples in this paper interprets A as an action predicate. It should be noted that this general approach to the treatment of action does not take into consideration state change and temporal dimension and is focused only on the agent concerned and the states of affairs that result from his/her actions. It is not a drawback as far as this work is concerned because when specifying rules/policies for normative co-ordination in an organisational setting where multiple actors are involved, it may be that only the end result together with which actors brings it about is important.

One other axiom advanced in (Santos et al. 1997) – and adopted here– to characterise specifically the action operator E is

$$E_xE_yA \rightarrow \neg E_xA \tag{2}$$

which corresponds to the idea that the brings-it-about operator expresses actions performed directly and personally. Compared to E_xA which pertains to *individual* agent positions E_xE_yA denotes interpersonal control positions and it is this ability to iterate action operators that could be seen as a benefit for having a general theory of action. From a process point of view the above axiom states a principle of rationality for modelling co-ordination between different actors in a process model. For instance, such an axiom can be used to show that an actor x delegates an actor y to bring about a condition. It is counterintuitive that the same agent brings it about that A and brings it about that somebody else achieves A.

2.2 Obligation

Our logical framework incorporates obligations. We use O as a directed deontic operator indexed by a set of agents to represent obligation. We write $O_xE_{\{y\}}A$ to mean that agent y has towards agent x the obligation of realising A. As in the case of E we need to sketch a suitable axiomatisation for O. We cannot use Standard Deontic Logic (SDL) for this purpose as it has been shown in (Royakkers 2000) that SDL is not adequate for combining deontic and action operators. The reason is that SDL supports the following implications which is not acceptable from a process modelling view point

$$O_yE_xA \rightarrow O_yA \tag{3}$$

$$O_zE_xE_yA \rightarrow O_zE_yA \tag{4}$$

For instance, let x be a retailer, y a customer and A is the condition *sendGoods*. Then the retailers obligation towards the customer to bring about the condition *sendGoods* should not entail that *sendGoods* is in general obligatory. Similarly in the case of the second implication suppose that we have a scenario where in addition to the retailer and customer a manufacturer z is also involved and A is the condition *buyProduct*. Now the retailers obligation towards the manufacturer to bring about the condition that the customer buys a particular product does not entail that the customer has a personal obligation to the manufacturer to buy that product. To avoid the above problems we only consider the following axioms for a logic of obligation

$$(O_xA \wedge O_xB) \rightarrow O_x(A \wedge B) \tag{5}$$

$$O_xA \rightarrow \neg O_x\neg A \tag{6}$$

2.3 Proclamation

The link between speech acts theory and normative positions has been under investigation for some time now (cf. (Jones 1990, Castelfranchi, Dignum, Catholijn & Treur 2000, Singh 1999, Colombetti 2000)). (Gelati, Rotolo, Sartor & Governatori 2004) defines proclamation as a special type of speech act (communication act) dealing with all those acts by which a subject makes a statement expressing a certain proposition and this statement has the function of making this proposition true. In this way it can be seen as a *see-to-it-that* modality indirectly representing a speech act and can be formalised by the modal operator *proc*. As for E, *proc* will be indexed by sets of agents and therefore $proc_XA$ means that the members of x jointly proclaim A. As before when X has only one element x, $proc_XA$ means that A is proclaimed personally by x. Its logic is characterised by some very minimal properties: it is closed under logical equivalence, i.e., $A \equiv B \rightarrow procA \equiv procB$ and includes at least the axiom

$$(proc_xA \wedge proc_xB) \equiv proc_x(A \wedge B). \tag{7}$$

Of course, *proc* is not necessarily successful. Whether it is successful or not, within an organisational setting s, depends on whether s makes it effective by means of appropriate *counts-as* rules. We talk more about the counts-as rule in the next section.

Since we define commitments in terms of directed obligation and commitments typically arise from certain communication acts, we can use *proc* to model a communication act through which a particular agent conveys his/her obligation towards another. For example in the scenario depicted in Figure1 $proc_r(O_tE_{\{r\}}A)$, conveys a communication act made by the retailer (r) meaning that the retailer is obligated towards the technician (t) to bring about a certain condition, for instance to pay a specific amount of money. This proclamation by r could be seen as r's attempt to commit itself towards t. In a similar manner to model a communication act expressing a joint commitment between two parties involved we can use $proc_{\{x,y\}}$. We discuss more about joint commitments in the coming sections. Also, as we have observed, proclamations are not necessarily effective in the sense that when an agent x proclaims that A, x brings it about that A is dependent on the concerned organisation. In the next section we show how we can achieve this result.

2.4 The *counts-as* Rule

In the previous sections we described the main ingredients needed to develop a theory of institutional agency. The intuition behind such an exercise was to show how agent-oriented approaches to normative agency can be used in the domain of process modelling/management. But we need some more material to complete the picture. We need a way to express that certain facts hold in the context of an institution. For instance, it is normal in a norm-governed institution that designated agents are empowered to create

institutional facts[2] by performing certain types of actions. Hence each organisation needs rules about which agents are empowered to assign rights, or to alter existing rights. Such power assignment rules can often be represented using a *counts-as* structure to denote the context in which they operate. In a business process scenario this is important as the context denotes a group that contains the participating agents usually in different roles thereby enabling the user to specify those rules that *count as* effective in a particular context. We represent this contextual structure using a conditional connective, $\Rightarrow_s$, to express the *counts as* connection in an institution s. It should be noted that this conditional connective is used just as a symbol of representation and has got nothing to do with the formalism developed in Jones and Sergot (Jones & Sergot 1996). The intuition behind the above definition of the counts-as link is that we want to capture the idea that counts-as rules may specify when an institutional act (e.g., a contract made by person x in the name of person y) has the same effects of another institutional act (e.g., a contract made by y).

Another nice feature of the counts-as rule is in its ability to express various forms of *normative delegation* when combined with *proc*. Since *proc* is not successful, its effectiveness is provided by the institution assuming rules such as

$$proc_x A \Rightarrow_s E_x A. \tag{8}$$

Such a combination can be used to capture two forms of delegation such as

$$proc_y(proc_x A) \Rightarrow_s E_y(proc_x A) \tag{9}$$

$$proc_y(E_x A) \Rightarrow_s E_y(E_x A) \tag{10}$$

where the proclamation or the action of y count as the proclamation or the action of somebody else. (9) conveys the meaning that when y proclaims that x proclaims that A, this counts as y making so that x proclaims that A (here x is the principal and y is the representative). For example, *Determine Warranty type* in Figure(1) is an automated activity which interprets the conditions of the after sales service agreement and appropriately routes the subsequent activities to the correct process role (retailer or manufacturer). There can be a condition in the agreement which states that if the purchase is made in the last 12 months, then the warranty will be covered by the manufacturer. However, the retailer provides extended warranty services, and if the problem is reported in the second or third year of the purchase, then the retailer will provide the maintenance service. Consider a particular situation in our scenario where the retailer r represents the manufacturer m with respect to informing the customer c that *if the purchase is made in the last 12 months then the warranty will be covered by the manufacturer*. This could be seen as a proclamation from the manufacturer denoting a business policy he/she follows to do business. We can formally define this policy as

$$proc_m(O_c E_m(coverWarranty(1^{st} year))) \tag{11}$$

where $(O_c E_m(coverWarranty(1^{st} year)))$ is the content of the manufacturer's proclamation denoting his/her obligation to the customer to cover warranty for the first year. Hence (9) can be reformulated as

$$proc_r(proc_m(O_c E_m(coverWarranty(1^{st} year)))) \Rightarrow_s E_r(proc_m(O_c E_m(coverWarranty(1^{st} year)))) \tag{12}$$

Therefore as far as c is concerned with the reading that r's proclamation about m's business policy that c's warranty will be covered by m for the first 12 months counts as r's making so that m proclaims the policy.

[2]For a distinction between *social facts* and *institutional facts* refer (Taveter & Wagner 2001*a*).

Let us see how such rules of delegation gets used in our scenario. Suppose that (12) is a business rule[3] representing certain conditions of the after sales service agreement. Once *Create Customer Request*, Figure(1), is performed by the customer using the designated service available through the portal it is up-to *Determine Warranty Type* to interpret conditions of the after sales service agreement and route subsequent activities accordingly. We will see how the conditions/constraints in rules like (12) gets interpreted/reasoned about. From $E_r(proc_m(O_c E_m(coverWarranty(1^{st} year))))$ (the consequent of (12)) and (1) we get

$$(proc_m(O_c E_m(coverWarranty(1^{st} year))). \tag{13}$$

Similarly from (13) and (8) we get

$$E_m(O_c E_m(coverWarranty(1^{st} year))) \tag{14}$$

Again by applying (1) to (14) we get

$$O_c E_m(coverWarranty(1^{st} year)) \tag{15}$$

conveying that the manufacturer is obliged to the customer to cover warranty for the first year and accordingly *Determine Warranty Type* will route further activities to the manufacturer if the customer request states that his/her purchase was done in the last 12 months. More complex constraints can be easily added to such rules as we show later in this paper. Also, our choice of the consequent part of (12) was arbitrary as we would get the same result with the antecedent too.

A representation like (10) is necessary when the representative substitutes a principal which would not be able to perform directly the activity delegated to the representative. Also when applied to action descriptions, formulas like

$$E_x A \Rightarrow_s E_x B \tag{16}$$

$$E_x A \Rightarrow_s E_y B \tag{17}$$

represent respectively x's institutional power to produce B when A is realised and x's power to perform an action as if something else were made by y (see (Jones & Sergot 1996)).

The last notion we have to deal with is that of *Declarative power*. The concept of declarative power is common in many normative systems and consists in the capacity of the power-holder of creating institutional facts, simply by "proclaiming" them. But as pointed out earlier, proclamations are not necessarily effective and when an institution provides for the effectiveness of a proclamation we say that the subject of the proclamation has a declarative power. The following definition holds

$$DeclPow_{\{x\}} A =_{def} proc_{\{x\}} A \Rightarrow E_{\{x\}} A \tag{18}$$

conveying the meaning that an agent x has the declarative power of producing A means that if x proclaims that A then x produces A. In a similar manner to show that every couple of actors has the power of establishing any obligation between them simply by proclaiming it we have the following representation

$$DeclPow_{\{x,y\}}(O_y E_{\{x\}} A) \tag{19}$$

3 Commitments via Proclamation, Obligation and Direct Action

In the previous sections we outlined various constructs needed for a theory of normative agency. In this section

[3]Business rules are statements that express (certain parts of) a business policy, defining business terms, and defining or constraining the operations of an enterprise (Taveter & Wagner 2001*a*).

we show how combining these different concepts we can arrive at a notion of commitment which is needed to effectively form a contract between two participating entities in a business scenario. In (Singh 1999, Yolum & Singh 2004) commitment is treated as first-class abstract objects where a base level commitment is represented as a four-place relation, $C(x,y,G,p)$, denoting a commitment from x toward y to bring about a condition p in the context of G. Here x is the agent who is committed (*debtor*) and y is the agent who receives the commitment (*creditor*). We could express this in our formalisation by a two step process without taking commitment as a primitive as follows,

$$O_y E_{\{x\}} p \tag{20}$$

with the reading that *x is obliged to y to bring it about p*. In order to show the commitment aspect we combine *proc* with the directed deontic operators above to give us

$$proc_{\{x\}}(O_y E_{\{x\}} p) \tag{21}$$

Here, the proclamation is x's attempt to commit itself towards y and thereby makes x *responsible* to y for satisfying p. For a stronger version of commitment we need to provide (21) with some additional support. It has been pointed out in (Gelati et al. 2004) that proclamations are not necessarily effective in the sense that when an agent x proclaims that p, x brings it about that p only if the institution provides for this result. But this is not a problem in our set up as we can give a rule like

$$proc_{\{x\}} p \Rightarrow_{SCM} E_{\{x\}} p \tag{22}$$

which conveys the meaning that in the context of Supply Chain Management (SCM) x's proclamation of p counts as x brings it about p. For example, let x denote a *supplier*, y a *customer* and p denote a condition like $sendGoods(p)$. Then from (21) and (22) we get the reading that, in the context of an SCM, the supplier's proclamation regarding his/her obligation towards the customer to bring about the act of sending goods counts as the supplier bringing about the act. In other words the supplier's proclamation regarding his/her commitment towards the customer counts as the supplier realising those commitments. Hence the supplier's commitment towards the customer to send goods results in sending the goods. This formalisation also goes well with the commonsensical view that commitments to other agents represent commitments to oneself to bring it about. In a business process scenario like SCM, (21) and (22) could be seen as *policies* that govern the commitment operations among different stakeholders and are part of the contractual relationships existing between them. They are considered as policies as they differ from the local business rules and configurations that make up a particular partner in a business scenario. In a similar manner (23) shows x's attempt to *command* y and (24) conveys x's attempt to *free* itself from an obligation towards y.

$$proc_{\{x\}}(O_x E_{\{y\}} p) \tag{23}$$

$$proc_{\{x\}}(\neg O_y E_{\{x\}} p) \tag{24}$$

3.1 Commitments Through Reciprocal Obligations

In the previous section we saw how a proclamation by a single agent, where a combination of directed obligation and action is involved, could account for base-level commitments of the type $C(x,y,G,p)$. But as pointed out in (Gelati et al. 2004) there could be *multi-lateral proclaims* within an institution where a set of agents is involved in a proclamation with reciprocal obligations as the content of the proclamation. We use such proclamations to substitute the conditional commitments as proposed in (Singh 1999). A conditional commitment $CC(x,y,p,q)$ denotes that if the condition p is satisfied, x will be committed to bring about condition q. In other words conditional commitments are useful when an agent wants to commit only if a certain condition holds or only if the other party is also willing to make a commitment. In our case this condition is achieved through a notion of mutual obligation. It is also the case that such proclamations can be used to denote meta-commitments which in turn are rules that govern the commitment operations as was pointed out in (Singh 1999). Reciprocal obligations are used as the content of such proclamations through which we capture the meta-commitment idea. Actually, in a business process scenario like SCM such joint proclamations carry more sense. For instance, in an exchange relationship between a supplier (x) and a customer (y), we can define a commitment between the two parties through a joint proclamation by combining (21) and (23) as follows;

$$proc_{\{x,y\}}(O_y E_{\{x\}}(sendGoods(p)) \wedge O_x E_{\{y\}}(sendMoney(q))) \tag{25}$$

The proclamation made in (25) shows the joint commitment between x and y by taking x's obligation towards y to send goods p and y's obligation towards x to send money q. In other words, there should exist reciprocal obligations between x and y to create such mutual commitments. Elsewhere (Gelati et al. 2004) the term *contract* is given for such commitments. It is also the case that a joint proclamation like (25) boils down to two further committing acts of *offer* and *accept* where x's offer to y is based on reciprocal obligations between x and y and y accepts this. For instance if x's offer to y is based on the reciprocal obligation between x and y that x sendGoods(p) and y sendMoney(q) and y accepts to it then this counts as making the commitment. Formally,

$$\begin{aligned} &offer_{\{x\},\{y\}}(sendGoods(p), sendMoney(q)) \wedge \\ &accept_{\{x\},\{y\}}(sendGoods(p), sendMoney(q)) \Rightarrow_{SCM} \\ &proc_{\{x,\{y\}}(O_y E_{\{x\}} sendGoods(p) \wedge \\ &O_x E_{\{y\}} sendMoney(q)) \end{aligned} \tag{26}$$

In this way we restrict commitments to the creation of reciprocal obligations. Now as noted in (Gelati et al. 2004) it is possible to define *offer* and *accept* which are basically committing acts in terms of non committing acts like *proposal* and *agree*. For instance

$$\begin{aligned} &proposal_{\{x\},\{y\}}(sendGoods(p), sendMoney(q)) =_{def} \\ &proc_x(O_y E_{\{x\}} sendGoods(p) \wedge \\ &O_x E_{\{y\}} sendMoney(q)) \end{aligned} \tag{27}$$

conveys x's declaration whereby he/she proposes not only to have an obligation towards y to do $sendGoods(p)$ but also to command y to do $sendMoney(q)$ (i.e., (21) and (23)). In similar lines we can show that y *agree* with x when x has already made a proclamation in which a specific contractual content is proposed for y to bring about and y makes a proclamation to commit itself towards x to bring about this content.

$$\begin{aligned} &agree_{\{y\},\{x\}}(sendGoods(p), sendMoney(q)) = \\ &proposal_{\{x\},\{y\}}(sendGoods(p), sendMoney(q)) \wedge \\ &proc_{\{y\}}(O_x E_{\{y\}} sendMoney(q)) \end{aligned} \tag{28}$$

From the above discussion we can say that an *offer* takes place in a business process scenario like SCM when the

following condition is satisfied;

$$offer_{\{x\},\{y\}}(sendGoods(p), sendMoney(q)) = proposal_{\{x\},\{y\}}(sendGoods(p), sendMoney(q)) \wedge \{agree_{\{y\},\{x\}}, (sendGoods(p), sendMoney(q)) \Rightarrow_{SCM} proc_{\{x,y\}}(O_y E_{\{x\}} sendGoods(p) \wedge O_x E_{\{y\}} sendMoney(q))\} \quad (29)$$

The main idea of (29) is to show how the contractual relationships among the participants (in this case its between the supplier and customer) in a business process scenario like SCM evolve over the course of an interaction. For instance, x's proposal to y of a specific contractual content and y's acceptance of it would create the respective directed obligations between them which in turn leads them to form mutual obligations and thereby arrive at a joint commitment. In other words (29) can be seen as a business policy stating that an offer is made between two stakeholders when there is a proposal and agreement between the two wherein there is mutual commitment regarding a specific contractual content. Such business rules help in outlining the guidelines and restrictions with respect to states and processes in an organisation. They are declarative statements describing *what* has to be done rather than *how* to do it. In a similar manner y's acceptance with regard to a contractual relationship could be given as

$$accept_{\{y\},\{x\}}(sendGoods(p), sendMoney(q)) = offer_{\{x\},\{y\}}(sendGoods(p), sendMoney(q)) \wedge proc_{\{y\}}(O_x E_{\{y\}} sendMoney(q)) \quad (30)$$

indicating y's agreement with the contractual content. In this manner, by applying a small set of operations like proposal, accept etc. on the combination of directed obligation and action we can represent various contractual relationships of interest in a business scenario. Table 3.1 shows that all the operations defined in (Singh 1999) can be captured in our formalism and some of them in a more intuitive manner. Except for *Assign* all the other operations have a direct reading. The idea of assign is to show that the holder x of the recursive declarative power can exercise his/her power in two ways. The first conjunct shows x's command over y so that y is obliged to realise A. The second conjunct enables x to transfer to another agent z the same declarative power x possesses i.e., Assign transfers a commitment to another creditor within the same context, and can be performed by the present creditor because it is authorised.

4 Bringing them all together

Now we will show how the formal representation developed above can be used to capture the normative dependencies involved in a scenario like the after sales process model as shown in Figure 1. Consider the part where the retailer/manufacturer sends notification to the technician. As noted earlier this may involve an internal activity like finding the most appropriate technician which in turn depends on certain agreements reached between a particular technician and the retailer/manufacturer. Let us capture this scenario using our framework. It should be noted that in this paper we are interested in the *committed dependency* between the actors i.e., in a committed dependency the dependee will try its best to perform the task because of the fact that the depender would be hurt significantly if the dependency fails [4]. Since in our framework we have a stronger version of committed dependency in the form of reciprocal obligations (joint commitments) we always have $O_y E_{\{x\}}(X) \wedge O_x E_{\{y\}}(Y)$ denoting the content of the commitment as was given in (25). In the case of our after sales service scenario which includes the retailer and a technician we could state this condition precisely as

$$O_r E_{\{t\}} performed(j) \wedge O_t E_{\{R\}} paid(p) \quad (31)$$

with the reading that the technician t undertakes toward the retailer r, the obligation to perform the job j, while the retailer r undertakes towards the technician t, the obligation of paying the price p. Further, a call for proposal (of making a commitment having content X) by retailer r from any technician $t \in T$ can be represented as $proposal_{\{r\},\{t\}}(X)$. In a similar manner $offer_{\{t\},\{r\}}(X)$ conveys technician t's offer to retailer r with respect to a commitment having content X and $accept_{\{r\},\{t\}}(X)$ means that the retailer r accepts to a commitment having content X with technician t.

Suppose that the retailer issues a proposal the terms of which states the reciprocal obligations of both parties involved. For instance, the retailers proposal could be that the technician has an obligation to repair a *LG washing machine (j)* and the retailer has the obligation to pay 60 Dollars (p) for it. This could be represented as

$$proposal_r^T(E_{\{t\}} performed(j), E_r paid(p)) \quad (32)$$

From (3.1) we can arrive at the conclusion that (32) refers to the retailers proclamation of a specific proposal. Though this inference is not of much use for this work it is useful when we think about message passing as a kind of speech act. The consequence of (32) is that those technicians who are capable of repairing LG washing machine can make offers. Suppose that one of the technicians returns an offer which is in the form of a counter-proposal.

$$offer_{\{t\},\{r\}}(E_{\{t\}} performed(j), paid(p')) \quad (33)$$

where $p' = 50$ Dollars. Assume that this is the best offer the retailer has received and therefore he/she accepts it. Now the acceptance by the retailer implies his/her agreement because by (29) an *offer* happens when a proposal and agreement is already in place. The acceptance by r could be given as

$$accept_{\{r\},\{t\}}(E_{\{t\}} performed(t), E_{\{r\}} paid(p')) \quad (34)$$

Using (33) and (34) along with (3.1) gives us

$$(proc_{\{r,t\}}(O_r E_{\{t\}} performed(t) \wedge O_t E_{\{r\}} paid(p'))) \quad (35)$$

which shows that the parties have made a joint commitment within the context of *After Sales Service Processing* (ASSP) scenario. Also because of (7), (35) implies

$$proc_{\{r,t\}}(O_r E_{\{t\}} performed(t)) \wedge proc_{\{r,t\}}(O_t E_{\{r\}} paid(p'))) \quad (36)$$

By applying the rules provided earlier which shows the effectiveness of a proclamation we can derive from (36) the conclusion that t is obliged to do the job and r is obliged to pay for it. Formally this is given as follows

$$(O_r E_{\{t\}} performed(t) \wedge O_t E_{\{r\}} paid(p')) \quad (37)$$

The above example tells us that agent's behaviour within organisations are governed by social rules that impose obligations over the agents actions. Therefore coordination in organisations and societies cannot be accounted for without considering the social laws of the organisations and the way they constrain behaviours of individual agents. Hence as a first step to make this view practically usable in applications it is important to represent and reason about the obliged behaviours within agents as we have shown above. This in turn would help to explain co-ordination among agents as negotiation about obliged behaviours.

[4] In an *open dependency* if the dependency fails the depender would be affected to some extent whereas a *critical dependency* indicates that some goal of the depender could not be achieved if the dependency fails. For an overview of this classification refer to (Yu & Mylopoulos 1993).

Operation	Meaning	Representation
Create	Instatantiates a commitment	$proc_{\{x\}}(O_y E_{\{x\}} A)$
Cancel	Revokes the commitment	$proc_{\{x\}}(\neg O_y E_{\{x\}} A)$
Release	Eliminates the Commitment	$proc_{\{x\}}(\neg O_x E_{\{y\}} A) \wedge proc_{\{x\}}(\neg O_x \neg E_{\{y\}} A)$
Delegate	Shifts the role of x (debtor) to another agent within the same context	$proc_{\{x\}}(proc_y A) \Rightarrow_s E_x(proc_y A)$ $proc_x(E_y A) \Rightarrow_s E_x(E_y A)$
Assign	Transfers a commitment to another creditor within the same context	$RecDecPow_{\{x\}}(O_x E_{\{y\}} A) =$ $DeclPow_{\{x\}}(O_x E_{\{y\}} A) \wedge$ $DeclPow_{\{x\}}(RecDeclPow_{\{z\}}(O_x E_{\{y\}} A)$

Table 1: Operations on commitments

5 Rules for Deontic Dependency

In this section we provide some general rules that takes care of the deontic constraints to be satisfied between actors. These rules in turn regulates the agent behaviour in an organisational set up. A rule stating the deontic constraints that need to be satisfied to make a proposal between a debtor (x) and creditor (y) can be stated as follows;

$$\begin{aligned} IF \quad & isDebtor(x) \\ & AND \quad isCreditor(y) \\ & \quad AND \quad (O_y E_{\{x\}} sendGoods(p)) \\ & \quad AND \quad (O_x E_{\{y\}} sendMoney(q)) \\ THEN \quad & proposal_{\{x\},\{y\}} \end{aligned}$$

In a scenario like in (1), we can use the above rule to represent the constraints related to a proposal of a contract issued by the Retailer. For instance, the retailers proposal could be that the technician has an obligation to *repairLG washingmachine*(q) and the retailer has the obligation to pay 60 Dollars(p) for it which could be given in the following way;

$$\begin{aligned} IF \quad & isRetailer(r) \\ & AND \quad isTechnician(t) \\ & \quad AND \quad (O_r E_{\{t\}} performed(q)) \\ & \quad AND \quad (O_t E_{\{r\}} paid(p)) \\ THEN \quad & proposal_{\{r\},\{t\}} \end{aligned}$$

Similarly, we can give a rule stating the constraints to be satisfied so that an *agreement* could be reached between retailer r and technician t regarding a specific contractual content as follows;

$$\begin{aligned} IF \quad & proposal_{\{r\},\{t\}}(performed(q), paid(p)) \\ & AND \quad proc_{\{t\}}(O_r E_{\{t\}} performed(q)) \\ THEN \quad & agree_{\{t\},\{r\}}(performed(q), paid(p)) \end{aligned}$$

i.e., in order for the retailer and the technician to come up with an agreement, initially, there should be a proposal from the retailer to the technician regarding a specific contractual content and the technician makes a proclamation through which he obliges himself to bring about the specific content. It is also possible that the technicain t can come up with a better *offer* for the retailer by quoting a different price p' for the job to be performed. A rule for an offer could be given as follows;

$$\begin{aligned} IF \quad & proposal_{\{t\},\{r\}}(O_r E_{\{t\}} performed(q)) \\ & AND \quad (O_t E_{\{r\}} paid(p')) \\ & \quad AND \quad agree_{\{r\},\{t\}}(performed(q), paid(p')i) \\ THEN \quad & offer_{\{t\},\{r\}} \end{aligned}$$

6 Accommodating Deontic Dependencies in i^*

In the previous sections we developed a framework that provides a normative description of a (business) process

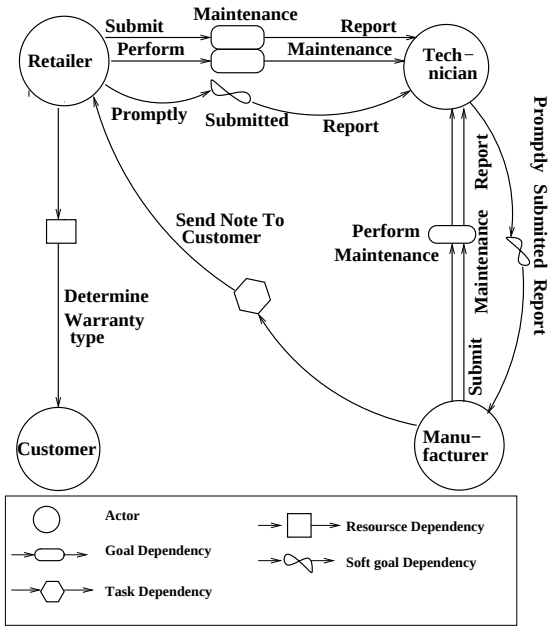

Figure 2: **SD**-model for the *After Sales Service Scenario* in Figure 1:

in their organisational settings. In this section we show a dependency diagram similar to i^* in which we can accomodate normative dependency. The i^* Framework (Yu & Mylopoulos 1994, Yu & Mylopoulos 1993), (pronounced i-star and stands for *distributed intentionality*), is an organisational modelling technique used by many groups around the world in their research on early requirements engineering, business process design, software development methodologies and many more. The i^* approach enables to describe, model and reason about the goals of systems (business and socio-technical) that involve many different actors and for choosing system architectures that best meet these goals. By explicitly modelling and analysing strategic relationships among multiple actors the approach incorporates rudimentary social analysis into a system analysis and design framework. The framework is based on the **SD** (*Strategic Dependency*) model and the **SR** (*Strategic Rationale*) model wherein the actors are related to each other *intentionally*. Actors depend on each other for goals to be achieved, tasks to be performed and resources to be furnished. Whereas SD-model is used to represent a particular design for a business process the SR-model describes the reasoning that actors have about the different possible ways of organising work, i.e, different configurations of SD networks. Since our main aim is to account for a notion of deontic dependency in the i^* framework we restrict ourselves to the SD-model.

Figure (2) shows a Strategic Dependency model for the after sales service work-flow in Figure (1). As can be seen from the figure the SD model consists of a set of nodes and links. Each node represents an actor, and each link

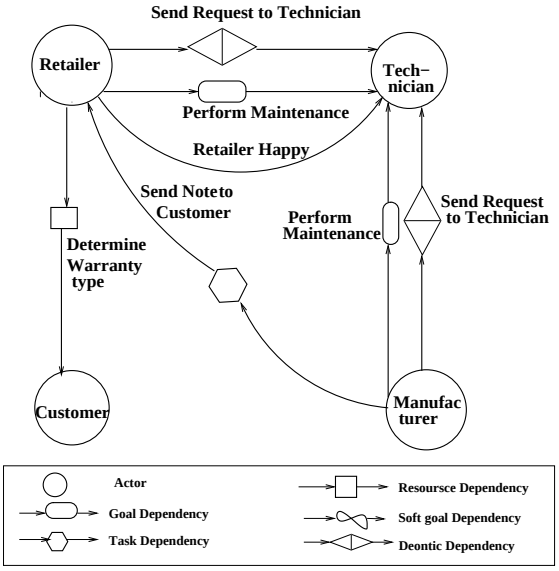

Figure 3: **SD**-model with Deontic Dependencies for the *After Sales Service Scenario* in Figure 1:

between two actors indicates that one actor depends on the other for something in order that the former may attain some goal. Four types of dependencies are distinguished in an SD-model between a **depender** (the depending actor) and a **dependee** (actor who is depended upon). The object around which the dependency relationship centres is called the **dependum**. In a *goal-dependency* a depender depends on the dependee to bring about a certain state in the world wherein the dependee is free to choose *how* to accomplish the goal. For instance, in Figure (1), the *Retailer/manufacturer* depends on the *Technician* in-order to perform maintenance. The retailer/manufacturer is only concerned about the outcome of *perform maintenance* and doesn't care how the technician achieves the goal. Another goal dependency in Figure (1) is that of *submit maintenance report* between the retailer/manufacturer and the technician. Similarly, in a *task-dependency*, though a depender depends on the dependee to carry out an activity (the dependum), the activity specification constrains the choices of the dependee on *how* the task is to be performed. As an example of task dependency, the *Manufacturer* depends on the *Retailer* to send notification to the customer regarding the warranty. This is so because the *Manufacturer* wants to have the warranty determined (limited/extended) according to the well defined instructions outlined by it. In a *resource dependency*, the depender depends on the dependee for the availability of an entity (physical or informational). The dependency between the *Retailer* and the *Customer* to determine warranty type is a resource dependency where the dependum is a piece of information related to the date on which the customer purchased the particular product. In a*softgoal dependency* a depender depends on the dependee to bring about a condition in the world wherein the criteria is not precisely defined as in the case of hardgoal dependency. The dependee has a number of ways for achieving the goal and the depender indicates which combination of choices would sufficiently meet the desired subgoal. Its usually considered that a softgoal is *satisficed* rather than satisfied. In Figure (1) *promptly submitted report* is a softgoal dependency between the retailer/manufacturer and the technician. We will add one more dependency, deontic dependency, to this list as shown in Figure (3) so that the logical framework provided in the previous sections can be used to represent and reason about such dependencies. It should be noted that we avoided some relations in Figure (3) which is already explained in Figure (1) so as not to have the figures cluttered. *Send Request to Technician* is a deontic dependency between the retailer/manufacturer and technician because as mentioned earlier the retailer/manufacturer has the right to choose the most appropriate technician based on certain agreements they have reached as a result of mutual obligation. The dependum, i.e., the activity of sending work request cannot be reduced to any other dependency as it involves obliged behaviours.

Now we will show how to capture the intentional dependencies of the SD-model using our framework developed in the previous sections. In the SD-model the external actor relationships as outlined above are characterised in terms of more basic intentional concepts like belief, goal, ability and commitment. We represent the intentional dependencies as meta-commitments by using the notion of reciprocal obligation as follows; Let us consider a commitment, $c = C(x,y,G,p)$, as given in (Singh 1999) wherein c is base-level if p does not refer to any other commitment and c is a meta-commitment if p refers to a base-level commitment. In the case of i^* framework, for a goal dependency, the condition p is an assertion $achieve(g)$ representing the goal g to be achieved by an agent. For a task dependency on task t, p is $done(t)$ and for a resource dependency on resource r, p is $avail(r)$. In our model we can represent these dependencies in terms of reciprocal obligation of the depender towards the dependee to satisfy the condition p. For instance, in our model, in the case of multilateral proclamations (for example (25), the condition p (which is the content of the proclamation) is always defined in terms of directed obligations i.e. the satisfaction of condition p is based on reciprocal obligation between the depender and the dependee. This means the commitments arrived at by such proclamations are not at base-level but are meta-commitments since p involves the obligations of both parties involved which in turn leads to a joint-commitment. Hence in a way we can say that in our model we represent an intentional dependency as a meta-commitment/reciprocal obligation of the depender towards the dependee to create commitments to satisfy condition p. To give an example, suppose that *Perform Maintenance* in Figure(2) is a goal-dependency of the retailer on the technician. Then this dependency can be represented as a reciprocal obligation of the technician towards the retailer through which the technician has an obligation towards the retailer to create a directed obligation (upon receiving a request for repair) to achieve the goal *Perform Maintenance*. In a similar manner we can capture the other dependencies in terms of reciprocal obligations.

7 Conclusions and Future Work

We showed how a combination of notions related to *obligation*, *proclamation* and *direct action* accounts for a deontic theory of commitment which in turn can be used to model business processes in their organisational settings. We first outlined a logical framework, which is based on a multi-modal logic, to represent the various deontic concepts. A peculiar feature of our logical framework is in the use of *proclamation* as a unique speech act that can model all other speech acts that characterise an organisation. For instance, in most other approaches, what we modelled as proclamation is represented through different types of speech acts (commissives, permissives, agreements, etc.) where each one is characterised by its own specific semantics. We considered these differences as instances of just one speech act since as far as we are concerned the differences only pertain to the content which is proclaimed. Then we went on to demonstrate how a combination of obligation, proclamation and direct action can account for at-least two types of normative delegation. The most crucial part of this work was developed next, where it was shown how to achieve normative co-

ordination by imposing social constraints/rules in the form of mutual obligations among the agents/actors. One consequence of this approach is that it allows agents to talk to each other based on their sets of obliged behaviours and thereby have a clean approach to negotiation. Another consequence is that by stating them as deontic constraints the co-ordination among agents can be seen as an *exchange of deontic constraints*. Further, we formalised the various operations that can be performed on commitments based on the new framework. Finally, we compared our model with the i^* framework to show that the various dependency relationships can be explained in terms of our work.

An organisational model like ours can be used to capture, support and enforce social patterns of behaviour of business processes operating in open environments. Open societies need mechanisms to systematise, defend and recommend right and wrong behaviour which in turn can inspire trust into the agents/process that will join them. In our model we make use of obligation, commitment etc. as norms to describe such expected behaviour. Also our model is rich enough to cover wide range of contexts for agent interaction. From a workflow point of view, recent works (Russel, van der Aalst, ter Hofstede & Edmond 2005) show that there has been a shift of perspective from *Workflow Control Patterns* and *Workflow Data Patterns* to that of *Workflow Resource Patterns* where modelling of resources (human/non human) and their interaction is of prime importance. We believe that our framework can contribute much to support modelling in the organisational context in which a process operates.

Though we do not address any computational issues in this paper, work is in progress to develop a computational framework based on the logical intuitions we have described here. A computation model based on Defeasible logic has already been proposed in this regard. Defeasible logic has been developed by Nute (Nute 1987) with a particular concern about computational efficiency and developed over the years by (Maher & Governatori 1999, Antoniou, Billington, Governatori & Maher 2000). The reason being ease of implementation (Maher, Rock, Antoniou, Billignton & Miller 2001), flexibility (Antoniou et al. 2000) (it has a constructively defined and easy to use proof theory which allows us to capture a number of different intuitions of non-monotonicity) and it is efficient: it is possible to compute the complete set of consequences of a given theory in linear time (Maher 2001). Having such an inference mechanism allows an agent to deduce the logical consequences of given obligations as well as helps in resolving conflicts among obligations (two important directions in which our work could be extended). At the moment, we have provided two extensions of standard Defeasible Logic. The first incorporates the notions of "counts as" and agency, as described in this paper (Governatori & Rotolo 2003, Governatori, Rotolo & Sadiq 2004). The second combines agency, BDI concepts and obligations (Governatori & Rotolo 2004). Our future work will be devoted to developing a unique framework which is able to deal with the cognitive component (BDI concepts), agency, and normative notions ("counts as" and deontic operators).

Before closing down we want to mention some related works that is of importance to this document. (Gelati et al. 2004) is the starting point for this work. But the major difference is in our use of reciprocal obligations to capture the commitment aspect involved in agents. Also, we address our work from a process modelling point of view whereas (Gelati et al. 2004) is concerned with legal reasoning. The same reason applies to (Tan & Thoen 1998). Two other closely related works are (Taveter & Wagner 2001*a*, Taveter & Wagner 2001*b*). In those works too the concept of mutual obligation is absent as well as they do not use deontic logic to represent the different normative concepts. (Yolum & Singh 2004, Singh 1999) provided some insights to our approach. They represent commitments as first class abstract objects using a four place relation whereas we show how commitments arise as a result of the obliged behaviours between the different agents. (Yu & Mylopoulos 1994) and (Yu & Mylopoulos 1993) had a major influence on our work and as was shown in the discussion section our framework can accomodate the various dependency relationships outlined there. The advantages of having a fifth dependency in the form *deontic dependency* is worth investigating. (Barbuceanu et al. 1999) uses obligation, permissions and interdictions (OPIs) to reason about the behaviour of social agents. OPIs are modelled by reducing deontic logic to a particular type of dynamic logic and then constraint satisfaction techniques are used to infer consequences and solve conflicts among obligations and interdictions. We believe that this is a good approach for future work though it will not be possible to account for many normative relations that are defined in a deontic logical setting by reducing it to some type of dynamic logic. (Dignum, Vázquez-Salceda & Dignum 2004) is another work similar to ours written from a software engineering perspective.

Acknowledgements

This work was supported by the Australia Research Council under Discovery Project No. DP0558854 on "A Formal Approach to Resource Allocation in Web Service Oriented Composition in Open Marketplaces".

References

Antoniou, G., Billington, D., Governatori, G. & Maher, M. J. (2000), A flexible framework for defeasible logics, *in* 'Proc. American National Conference on Artificial Intelligence (AAAI-2000)', AAAI/MIT Press, Menlo Park, CA, pp. 401–405.

Barbuceanu, M., Gray, T. & Mankovski, S. (1999), 'Role of obligations in multiagent coordination', *Applied Artificial Intelligence* **13**(1-2), 11–38.

Castelfranchi, C., Dignum, F., Catholijn, M. & Treur, J. (2000), Deliberative normative agents: Principles and architecture, *in* 'Proceedings of ATAL 1999', Springer, Berlin, pp. 364–378.

Colombetti, M. (2000), A commitment-based approach to agent speech acts and conversations, *in* M. Greaves, F. Dignum, J. Bradshaw & B. Chaibdraa, eds, 'Proceedings of the Fourth International Conference on Autonomous Agents, Workshop on Agent Languages and Conversation Policies', Barcelona, pp. 21–29.

Conte, R. & Dellarocas, C. (2001), *Social Order in Multiagent Systems*, Kluwer Academic Publishers, Boston.

Dignum, V., Vázquez-Salceda, J. & Dignum, F. (2004), Omni: Introducing social structure, norms and ontologies into agent organisations., *in* 'PROMAS', Vol. 3346, Springer, pp. 181–198.

Elgesem, D. (1997), 'The modal logic of agency', *Nordic Journal of Philosophical Logic* **2**, 1–48.

Gelati, J., Rotolo, A., Sartor, G. & Governatori, G. (2004), ' Normative autonomy and normative co-ordination: Declarative power, representation and mandate', *Artificial Intelligence and Law* **12**, 53–81.

Goldkuhl, G. (1996), Generic business frameworks and action modelling, *in* 'Proceedings of the First International Workshop on Communication Modelling', Electronic workshops In Computing.

Governatori, G. & Rotolo, A. (2003), A defeasible logic of institutional agency, *in* 'Proceedings of the Fifth International Workshop on Nonmonotonic Reasoning, Action, and Change (NRAC'03)', pp. 97–104.

Governatori, G. & Rotolo, A. (2004), Defeasible logic: Agency, intention and obligation, *in* '7th International Workshop on Deontic Logic in Computer Science, DEON 2004', Springer, pp. 114–128.

Governatori, G., Rotolo, A. & Sadiq, S. (2004), A model of dynamic resource allocation in workflow systems, *in* 'Fifteenth Australasian Database Conference (ADC2004)', Australian Computer Science Association, pp. 197–206.

Jones, A. (1990), Towards a formal theory of communication and speech acts, *in* P. Cohen & M. Pollack, eds, 'Intentions in Communication', MIT Press, Cambridge, Mass.

Jones, A. & Sergot, M. (1996), 'A formal characterisation of institutionalised power', *Journal of IGPL* **3**, 427–443.

Kanger, S. (1972), 'Law and logic', *Theoria* **38**, 105–32.

Lindahl, L. (1977), *Position of change: A Study in law and logic*, Reidel, Dordrecht.

Maher, M. (2001), 'Propositional defeasible logic has linear complexity', *Theory and Practice of Logic Programming* **1**(6), 601–711.

Maher, M. J. & Governatori, G. (1999), A semantic decomposition of defeasible logic, *in* 'Proceedings of the 16th National Conference on Artificial Intelligence (AAAI-99', AAAI Press, pp. 299–305.

Maher, M. J., Rock, A., Antoniou, G., Billignton, D. & Miller, T. (2001), 'Efficient defeasible reasoning systems', *International Journal of Artificial Intelligence Tools* **10**(4), 483–501.

Nute, D. (1987), Defeasible logic, *in* 'Handbook of Logic in Artificial Intelligence and Logic Programming', Vol. 3, Oxford University Press, pp. 353–395.

Pitt, J. (2005), *Normative Specifications in Multi-Agent Systems*, John Wiley & Sons.

Pörn, I. (1977), *Action Theory and Social Science: Some Formal Models*, Reidel, Dordrecht.

Royakkers, L. (2000), Combining deontic and action logics for collective agency, *in* J. Breuker et al., ed., 'Legal Knowledge and Information Systems (Jurix)', IOS Press, Amsterdam.

Russel, N., van der Aalst, W. M. P., ter Hofstede, A. H. M. & Edmond, D. (2005), Workflow resource patterns: Identification, representation and tool support., *in* 'CAiSE', Vol. 3520, Springer, pp. 216–232.

Santos, F., Jones, A. & Carmo, J. (1997), Action concepts for describing organised interaction, *in* 'Thirtieth Annual Hawaii International Conference on System Sciences', IEEE Computer Society Press, Los Alamitos.

Singh, M. P. (1999), 'An ontology for commitments in multi-agent systems: Towards a unification of normative concepts', *Artificial Intelligence and Law* **7**, 97–113.

Singh, M. P. (2004), Business process management: A killer app for agents, *in* 'Proc. Autonomous Agents and Multi-Agent Systems (AAMAS-2004)', IEEE.

Singh, M. P., Chopra, A. K., Desai, N. & Mallya, A. U. (2004), Protocols for processes: Programming in the large for open systems (extended abstract), *in* 'OOPSLA Companion', Vol. 39, pp. 120–123.

Tan, Y.-H. & Thoen, W. (1998), Modelling directed obligations and permissions in trade contracts, *in* 'HICSS (5)', pp. 166–175.

Taveter, K. & Wagner, G. (2001*a*), Agent-oriented enterprise modelling based on business rules., *in* 'ER', Vol. 2224 of *Lecture Notes in computer Science*, Springer, pp. 527–540.

Taveter, K. & Wagner, G. (2001*b*), A multi-perspective methodology for modelling inter-enterprise business processes., *in* 'ER (Workshops))', Vol. 2465 of *Lecture Notes in computer Science*, Springer, pp. 403–416.

Wagner, G. (2003), 'The agent-object-relationship metamodel: towards a unified view of state and behavior.', *Inf. Syst.* **28**(5), 475–504.

Yolum, P. & Singh, M. P. (2004), 'Reasoning about commitments in the event calculus: An approach for specifying and executing protocols', *Annals of Mathematics and Artificial Intelligence. Special issue on computational Logic in Multi-Agent Systems.* **42**(4), 227–253.

Yu, E. S. K. & Mylopoulos, J. (1993), An actor dependency model of organizational work: with application to business process reengineering., *in* 'COOCS', ACM, pp. 258–268.

Yu, E. S. K. & Mylopoulos, J. (1994), From E-R to A-R - modelling strategic actor relationships for business process reengineering., *in* 'ER', Vol. 881 of *Lecture Notes in Computer Science*, Springer, pp. 548–565.