

"Figaro should be in Sydney by the 2nd of July" - Contracting in Many-to-Many E-services

Olivera Marjanovic^{*)} and Zoran Milosevic^{#)}

**)School of Information Systems, Technology and Management
University of New South Wales
Sydney, Australia
o.marjanovic@unsw.edu.au*

*#)Distributed Systems Technology Centre (DSTC)
Brisbane, Australia
zoran@dstc.edu.au*

Abstract: Dynamic e-business is the latest development in e-commerce that is based on a concept of many-to-many e-services where applications (services) can be wrapped and presented as independent e-services or composed to create new e-services. This paper investigates a problem of contract preparation in many-to-many e-services by combining temporal and deontic logic. Contract composition is illustrated by an example called "eBigMove".

1. INTRODUCTION

Dynamic e-business is the latest development in e-commerce that involves rapid teaming of companies with both familiar and new business partners in pursuit of specific business objectives. It is based on a concept of many-to-many e-services where applications (services), possibly offered by different companies, can be wrapped and presented as independent e-services that, in turn, could be further composed to create new e-services (Durante et al., 2000). Furthermore, to provide added value, it should be possible to customize and deploy composite e-services in a

very flexible and efficient way. Thus, it is not hard to imagine that composition of e-services poses a unique set of technical challenges such as coordination, security, data integrity, dynamic modification etc.

Currently, there are many companies that already offer or are in the process of developing technical platforms and solutions to support dynamic composition of e-services. For example, HP's E-speak platform (Kuno, 2000), Microsoft's Windows DNA (including BizTalk), IBM's San Francisco Framework etc. For more details these technical infrastructures see (Kuno, 2000). As technical components are becoming available, the IT community is shifting its interest from technical platform issues toward modeling of business interactions in composite e-services.

In order to ensure legality and protect interests of all parties involved in e-services, business interactions are regulated by contracts. Consequently, many-to-many e-services poses some interesting challenges in the area of e-contracting.

The main objective of this paper is to investigate a problem of contract preparation in many-to-many e-services. The proposed model is based on a combination of temporal and deontic logic used to express obligations, permissions and prohibitions as well as various temporal constraints and estimates. To illustrate contract composition including verification of its temporal and deontic consistency, we use a simple example of many-to-many e-services called "eBigMove".

2. MOTIVATING EXAMPLE

At the time of writing, one of the authors of this paper is about to move interstate. As all people who have ever moved their house know, the "big move" requires contacting a number of agents and service providers in order to move people, pets, furniture and cars and get them at their destination on time. Thus, it is necessary to arrange furniture shipment, transport of cars, reserve and buy plain tickets, organise a move for Figaro the cat etc.

The big move is, in fact, a scheduling problem constrained by a number of temporal constraints imposed both by service providers and customers. For example, different furniture removalists have different schedules and take different time to provide a service (e.g. one provides a service 4 times per week and on average takes 2-3 days to deliver furniture). Figaro the cat, should fly on the same flight as its owners and if that is not possible, it should not arrive before its owners. Cars should be delivered after the owners arrive, otherwise the price for this service will increase to include daily parking fee at the airport depot. Customers (and preferably their furniture and pets) should be in Sydney by 2nd of July.

All service providers are independent i.e. they are not interested (expected) to coordinate their activities in any way. Thus coordination involves a lot of phone calls and paper work as customers have to negotiate individual contracts with each

service provider, manually “compose” (i.e. schedule) their own “big move” and coordinate individual services.

Now imagine being able to find a service provider that is able to arrange a composite many-to-many e-service called “eBigMove”. So, the customers wouldn’t need to communicate with all individual service providers and manually coordinate their services. The provider of “eBigMove” would not only compose and coordinate individual services but also monitor their execution and, if necessary, replace one service with another one on the fly (e.g. in a very unlike case of airline strike).

Having in mind that different customers may have different requirements in terms of their temporal constraints, preferred service providers or priorities, this relatively simple problem of composition of individual services, can turn out to be very complex. More examples of complex many-to-many services can be found in construction industry (where a general constructor subcontracts a number of service providers from various specialty areas), film industry, telecommunication, software engineering etc. Note that, in general, companies forming coalition to pursue market opportunities are not a new concept. However, “the manual and tedious process required to form these coalitions limits the number of market opportunities that can be pursued” (Nayak et al. 2001, pg.2) as coalitions cannot be formed quickly enough to meet market demands. In the following sections of this paper we will concentrate on the problem of contract composition and illustrate how responsibilities and actions of individual service providers can be scheduled and coordinated.

3. E-CONTRACT BUILDING BLOCKS

3.1 The Reference Model of Open Distributed Processing

The reference model of open distributed processing (RM-ODP) (ISO/IEC, 1998) is increasingly being used for modeling of complex, open distributed systems. Its part called *the enterprise viewpoint* has been used as a practical framework for modeling of virtual enterprises, in particular e-contracts in B2B services (see for example Herring and Milosevic, 2001). In this section, we provide a brief overview of the basic concepts applicable to e-contracting.

A contract is an agreement governing part of the collective behaviour expressed in terms of roles and their responsibilities (obligations, permissions and prohibitions). An obligation is a prescription that a particular behaviour is required. An obligation is fulfilled by the occurrence of the prescribed behaviour. A permission is a prescription that a particular behaviour is allowed to occur. A permission is equivalent to there being no obligation for the behaviour not to occur.

A prohibition is a prescription that a particular behaviour must not occur. A prohibition is equivalent to there being an obligation for the behaviour not to occur. To formally model obligations, permissions and prohibitions we use a formal logic called deontic logic. This logic was introduced by von Wright (1968) and later widely applied to modeling of organizational knowledge: see for example: Lee (1988), ISO/IEC WD 15414 (1998), Cole et al. (2001) etc.

3.2 Modeling of Time

The ODP-RM Enterprise view is yet to address the temporal nature of obligations, permissions and prohibitions (Cole et al, 2001). However, proper modeling of temporal constraints is critical for many-to-many e-services especially for their selection, scheduling, monitoring and coordination. Modeling of time and temporal reasoning have been investigated for many years in many disciplines including artificial intelligence (see for example: (Allen, 1981), (McDermot, 1982), Dehter et. al, 1991), temporal databases (see for example: (Jajodia et al. 1998), software engineering (Manna & Pnueli, 1979) etc. However, in this section we will limit our discussion to the basic temporal concepts used in e-contracting.

- **Absolute time**

An absolute time value (also called time point) is commonly specified in terms of UTC (Universal Coordinated Time). A pair absolute of time values ($t1$, $t2$) such that $t1$ precedes $t2$ ($t1 \leq t2$) is called a time interval.

- **Relative time**

A concept of relative time is used to model time duration that is independent from any time point e.g. 2 days, 5 hours.

- **Repetitive (periodic) time**

A repetitive time is a set of ordered time points such that the distance between two consecutive time points is constant and corresponds to some relative time value d . Thus, a repetitive time value can be represented as:

$$r = (tb, te, d)$$

where tb and te correspond to the beginning and end of a time interval that represents the domain of the repetitive time while d is a relative time value.

- **Temporal constraints**

Temporal constraints are different rules that regulate the order, timing and duration of individual actions. *Hard* temporal constraints usually result in some consequences if the corresponding action is not performed as required (e.g. late grant applications are not accepted). Soft temporal constraints imply that the original temporal constraints could be relaxed under certain circumstances.

- **Notation**

The following notation is used for formal definitions of various constraints:

- *action-id* is a unique action identifier
- *temporal-operator* $\in \{<, \leq, =, >, \geq\}$ is used for comparison of either two relative time values or two absolute time values
- *d-limit* is a relative time value that corresponds to the prescribed time limit
- *type* $\in \{h,s\}$ determines the type of temporal constraint i.e. *h* corresponds to *hard* and *s* to *soft* temporal constraint.
- *temporal-reference* $\in \{‘b’, ‘e’\}$ is used to denote beginning ‘b’ or end ‘e’ of an action.
- *deadline* is an absolute time value e.g. *Date1*, *Date2* etc.
- *time-period* is a relative time value that determines the period of repetition of an action
- *b-time-point* and *e-time-point* two absolute time points that determine a domain of repetitive time
- *otime* denotes an absolute time value when an action is estimated to occur

The above notation should be used when interpreting the following definitions of temporal constraints.

- **Formal definition of temporal constraints**

A *duration constraint* limits *duration* of individual actions as follows:

Duration (action-id, temporal-operator, d-limit, type)

For example:

Duration (ai, \leq , d, h)

prescribes that action *ai* must take no more than *d* time.

Duration (ai, \geq , d, s)

prescribes that action *ai* should take no less than *d* time to complete.

An *absolute deadline constraint* limits in terms of absolute time, when an action must/should finish (e.g. the deadline for grant applications is 15.March, 2001, 5pm).

A_Deadline (action-id, temporal-reference, temporal-operator, deadline, type)

For example:

A_Deadline(ai, e, \leq , Date1, h)

prescribes that action *ai* must be completed no later than *Date1*.

A_Deadline(ai, b, \leq , Date1, s)

prescribes that action *ai* should start no later than *Date1*.

A *relative deadline constraint* limits when an action *must/should begin/end* relative to the *beginning/end* of another action. The distance between two reference points is expressed in terms of relative time. Formally:

R_Deadline(action1-id, temporal-reference, temporal-operator, action2-id, temporal reference, distance, type)

For example,

R_Deadline (aj, b, \leq , ai, e, d, h)

prescribes that action aj must start no later than d time after action ai is completed.

Note that relative deadline constraints can be also used to prescribe order of individual actions. For example,

$$R_Deadline (aj, b, =, ai, b, -, s)$$

prescribes that actions ai and aj should start at the same time.

Periodic deadlines are temporal constraints used to prescribe the occurrence of an action in terms of repetitive time. Formally,

$$P_Deadline (action-id, temporal\ reference, time-period, b-time-point, e-time-point, type)$$

For example:

$$P_Deadline (ai, e, d, Date1, Date2, h)$$

prescribes that action ai must be completed every d time starting from $Date1$ until $Date2$ is reached.

A set of temporal constraints is *mutually consistent*, if and only if it is possible to find any assignment of temporal attributes (beginning, end and duration) for all actions such that all temporal constraints can be satisfied.

- **Temporal estimates**

Temporal estimates describe estimated duration and order of individual actions. They are usually based on the accumulated experience and are important for scheduling of individual actions and resource planning. Formally,

$$EDuration (action-id, temporal-operator, d-limit)$$

For example:

$$EDuration (ai, =, d)$$

is interpreted that action ai could take (usually takes) d time to complete.

Estimated occurrence is used to express the fact that an action could occur after/before some absolute time or periodically every d time.

$$EOccurrence (action-id, temporal-reference, temporal-operator, otime)$$

For example:

$$EOccurrence (ai, b, <, Date1)$$

indicates that ai could start before $Date1$.

Estimated order is used to express how an action could start/end relative to the beginning/end of another action.

$$EOrder(action1-id, temporal-reference, temporal-operator, action2-id, temporal-reference)$$

For example:

$$EOrder(ai, b, <, aj, b)$$

is interpreted that action ai could start before action aj starts.

3.3 Deontic constraints

In role-based models (such as for example e-contracting), roles and their responsibilities have to be specified explicitly to prevent any possible misunderstanding or ambiguity. In terms of temporal attributes, a contract specification includes two temporal attributes: an absolute time indicating when the contract was signed and a time interval that specify the period of contract's validity. Formally, a contract can be specified as follows (note that for simplicity all other attributes are omitted):

$$C(\text{contract-id}, \dots, \text{date-signed}, \text{c-begin}, \text{c-end})$$

where c-begin and c-end determine the period of contract validity.

Now suppose that contract *ci* is signed on *Date1* and has a period of validity is (*cb*, *ce*).

$$C(\text{ci}, \dots, \text{Date1}, \text{cb}, \text{ce})$$

As already stated, a contract is formally defined as a set of deontic constraints assigned to various roles. Our representation of deontic constraints is based on deontic logic that is extended to include the concept of time.

- **Obligations**

An obligation can be formally represented as:

$$O(\text{role}, \text{action-id}, \text{temporal-reference}, \text{temporal-operator}, \text{deadline}, \text{tdistance}, \text{ob}, \text{oe})$$

where *role* is obliged to perform *action-id* either by the *Deadline* or every *tdistance* starting from *ob* until *oe* is reached. Note that (*ob*, *oe*) is the period of validity of this deontic constraint. This deontic constraint is properly defined if the following conditions are satisfied:

- Time interval (*ob*, *oe*) has to be contained within (*cb*, *ce*) i.e.

$$cb \leq ob \leq oe \leq ce$$

- Absolute time value *deadline* has to be within the period of validity of this deontic constraint i.e.

$$ob \leq \text{deadline} \leq oe$$

- In the case of repetitive time, Role must be able to perform *action* at least once i.

$$ob + \text{tdistance} \leq oe$$

The following are some examples of obligations:

$$O(r1, ai, e, \leq, \text{Date1}, -, t1, t2)$$

it prescribes that role *r1* is obliged to finish action *ai* no later than *Date1*. This obligation is valid from time *t1* to *t2*. Observe that *tdistance* attribute is not applicable to this type of deontic constraint. This deontic constraint will generate two temporal constraints as follows:

If *Date1* = *t2* then the deadline could not be extended and both generated temporal constraints will be hard:

$A\text{-Deadline}(a1, e, \leq, Date1, h)$

$A\text{-Deadline}(a1, b, >, t1, h)$

However, if $Date1 < t2$ then the first temporal constraint will become soft:

$A\text{-Deadline}(a1, e, \leq, Date1, s)$

- **Permissions**

A permission can be formally represented as:

$P(\text{role}, \text{action-id}, \text{temporal-reference}, \text{temporal-operator}, \text{deadline}, \text{tdistance}, \text{pb}, \text{pe})$

indicates that *role* is permitted to perform *action-id* either by the *deadline* or every *tdistance* starting from *pb* until *pe* is reached.

Similarly to obligations, a permission has to be valid during the period of contract's validity; absolute time value deadline has to be within the period of validity of this permission; and in a case of repetitive time, a role should be able to perform *action-id* at least once. For example:

$P(r1, ai, b, >, Date1, -, t1, t2)$

states that role *r1* is permitted to start action *ai* after *Date1* and it is valid from time *t1* to *t2*.

Permissions do not result in temporal constraints as they don't prescribe that action *ai* must occur. Rather, two temporal estimates will be generated as follows:

$EOccurrence(ai, b, >, Date1)$

$EOccurrence(ai, e, \leq, t2)$

meaning that action *ai* could be expected to start after *Date1* and finish by *t2*.

The following is an example of periodic permission:

$P(r2, ai, b, =, -, d, pb, pe)$

that can be interpreted as role *r2* is permitted to perform action *ai* every *d* time starting from *pb* until *pe* is reached. This will generate a number of temporal estimates:

$EOccurrence(ai, b, =, pb+d)$

$EOccurrence(ai, b, =, pb+2d)$

The number of temporal estimates is equal to the maximum number *n* such that:

$$pb + nd \leq pe$$

- **Prohibitions**

As already stated prohibitions are used to express that an action is forbidden to happen. Formally,

$F(\text{role}, \text{action-id}, \text{temporal-reference}, \text{temporal-operator}, \text{atime}, \text{fb}, \text{fe})$

states that role is forbidden to perform *action-id* during a certain period of time - that is determined by absolute time value *atime* and the period of validity of this deontic constraint: *fb* and *fe*. Note that prohibitions are defined for a period of time rather than repetitively. Similarly to permissions and obligations, this deontic constraint is properly defined if its period of validity is within the period of contract's validity and *atime* is within (*fb, fe*).

4. COMPOSITION OF AN E-CONTRACT IN MANY TO MANY E-SERVICES

In a simple contract, it is possible to start from deontic constraints and then schedule corresponding actions based on generated temporal constraints. However, in complex many-to-many services, the same process cannot be easily applied because it is very difficult to specify deontic constraints (in particular their temporal attributes) without prior scheduling of individual actions. Furthermore, complex e-contracts may involve a large number of service providers (each with different temporal constraints and estimates). Thus scheduling of individual services (i.e. their beginning and end times, their order and estimated durations) can be very challenging and can involve several iterations.

To start the scheduling process, we propose to use a time visualisation method called a time map (as depicted by Figure1). Nodes of this map are absolute time points that correspond to beginning/end time of individual actions. Arcs are relative time values that correspond to distance between time points (e.g. duration of an action). All arcs are labelled by temporal operators (e.g. “<”) and some by relative time values indicating time limits (“<d1”) – meaning that the distance between two time points should be less than d1. An absolute time value attached a node correspond to a deadline or estimated occurrence. To indicate repetitive time, a set of absolute time values is attached to a node. To distinguish temporal constraints from estimates, a darker font/colour is used.

Note that a time map constructed in this way may not contain all actions. The challenge of a proper scheduling is to find out how these additional actions can be linked with the rest of the time map in order to specify their order, duration and expected beginning/end times. As a solution to this problem we propose to modify and apply the Floyd-Warshal all pair shortest algorithm (Dechter et. al. 1991) that is used in artificial intelligence for temporal constraint networks. The algorithm has to be modified to take into account time estimates and repetitive time. The actual specification of this algorithm is out of the scope of this paper.

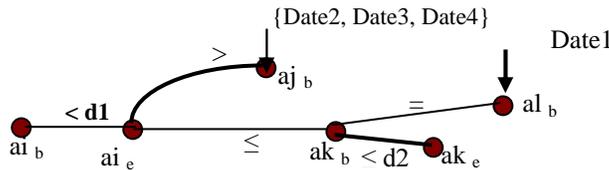


Figure 1. An example of a time map

Once when a possible schedule is found (as a result of this algorithm), it will be used for specification of deontic constraints for all service providers. The next step is to verify mutual consistency of the resulting deontic constraints. For example,

deontic inconsistency will arise when the same role is both obliged and forbidden, or both permitted and forbidden to do the same action at the same time. In other words, if a role is obliged to perform an action during a particular period of time, it has to be permitted to do it at the same time. Only when all deontic constraints are specified and verified, it is possible to formulate individual contracts for all service providers.

5. WILL FIGARO BE IN SYDNEY BY THE 2ND OF JULY?

To illustrate the introduced procedure of contract composition, let us go back to the example introduced in Section 2. So how to get Figaro the cat, its owners and preferably their furniture and cars to Sydney by the 2nd of July. The provider of eBigMove service will start from the required actions and temporal constraints specified by customers:

- Transport people by the 2nd of July (assume that this corresponds to action $a1$)
- Transport Figaro at the same time or after its owners by no later than the 2nd of July (action $a2$)
- Transport cars so that arrive in Sydney after its owners etc. (action $a3$)

Then the provider has to take into account various time estimates:

- A flight takes 1h.
- Car transport takes between 1 and 2 days and it is organised every 3 days etc.

These temporal constraints and estimates can be represented as:

$A_Deadline(a1, e, \leq, "2/7/2001: 19:00:00 +10", h)$
 $R_Deadline(a2, b, =, a1, b, 0, h)$
 $R_Deadline(a2, e, =, a1, e, 0, h)$
 $E_Order(a3, e, >, a1, e)$
 $EDuration(a3, \leq, 2days)$
 $EDuration(a1, =, 1h)$
 $EOccurrence(a3, b, "27/6/2001: 8:00:00 +10")$
 $EOccurrence(a3, b, "30/6/2001: 8:00:00 +10")$
 $EOccurrence(a3, b, "3/7/2001: 8:00:00 + 10")$

Then, the eBigMove provider will compose an initial time map (Figure 2).

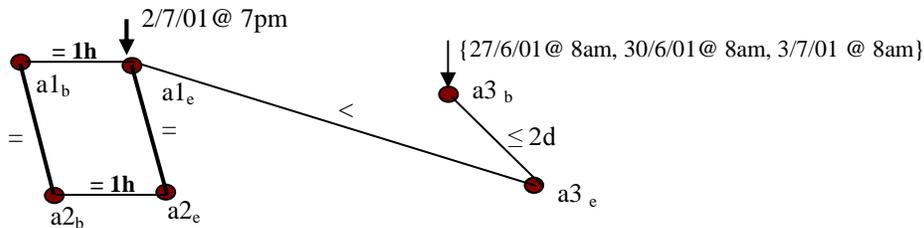


Figure 2: An initial time map for "eBigMove" composite e-service

Then, every time when a new service provider is selected, a set of new temporal constraints and estimates will be added to the existing time map. For example: A furniture removalist provides a service 4 times per week and usually takes 2-3 days to deliver it (Suppose that this action called “furniture delivery” is identified as $a4$). Furthermore, they require the customers to be present when furniture is packed for the transport (e.g. for insurance purposes), etc. This will result in an additional temporal constraint:

$$R_Deadline(a1, b, \geq, a4, b)$$

As a result, the two new nodes that correspond to $a4_b$ and $a4_e$ will be added to the map as well as a relative deadline constraint between $a1_b$ and $a4_b$.

Hence, scheduling of individual actions (services) is an interactive process where temporal constraints and estimates of potential service providers are removed, added and modified until a time schedule is found such that all temporal constraints are satisfied. Note that, the same schedule has to satisfy other constraints specified by the customer such as cost, quality etc. that are removed from this consideration for simplicity. In a case that a schedule cannot be found, then the customer is contacted and some of the initial temporal constraints are removed/modified and the whole process is repeated again.

When a schedule is found, the time map is then used to derive deontic constraints for all service providers. For example, the following obligation will be derived:

$$O(\text{Furniture-removalist}, a4, e, \leq, \text{“3/7/2001: 8:00:00 + 10”}, \text{“30/6/2001: 9:00:00 + 10”}, \text{“4/7/2001: 16:00:00 + 10”})$$

It indicates that a furniture removalist is obliged to deliver furniture preferably by 3rd of July at 8 a.m. but no later than the 4th of July 4pm and the contract was signed on the 30th of June at 8am.

When all deontic constraints are determined in this way and verified to prevent any possible inconsistency, the individual contracts can be signed. Note that the customer signs the contract only with eBigMove and they then sign a number of contracts with individual service providers. Thus, all these contracts are interrelated.

When all contracts are signed, the next step would be to monitor their execution and if necessary use the time map again to refine temporal constraints and fine tune execution of individual actions as well as coordinate the run-time replacement of one service provider with another. So will Figaro get to Sydney by the 2nd of July? Only if corresponding deontic constraints are satisfied!

After the same service is provided a number of times, the accumulated experience can be used to create new time estimates as well as to select the service providers that will satisfy customer requirements in the best possible way. However, monitoring of contract execution and analysis of the accumulated experience are out of the scope of this paper.

Although the previous example is relatively simple, it illustrates the complexity of contract composition and execution. Note that the introduced model can be further extended to deal with temporal constraints that are the function of different parameters e.g. price. In that case the scheduling algorithm has to be modified as well. This area is one of our future challenges.

6. RELATED WORK

A B2B Enterprise Model introduced by (Milosevic & Bond, 1995) is used as a basis of an enterprise model for many-to-many e-services. This model is currently being implemented using BizTalk technology and XML messaging (for more details see Herring & Milosevic, 2001). In order to support many-to-many e-contracting, we propose to extend the original model with a component called *contract verifier*. This decision support component needs to provide tools for construction and analysis of time maps, automatic scheduling of individual services (based on the Floyd-Warshal all pair shortest algorithm) as well as tools for user-friendly specification of deontic constraints and automatic verification of their consistency.

Other related work in the area of e-contracting includes EU-funded COSMOS project (see Griffel et al., 1998) that provides the set of services that facilitate the use of e-contracts. Much of the system deals with lower-level communication and representation issues rather than more contract-specific issues.

In the area many-to-many e-services, one of the leading research and development groups is certainly the HP group (see Durante et. al., 2000; Kuno, 2000). Their projects include development of technical architecture e-services as well as modeling and composition of e-services. However, they do not consider e-contracting aspects.

7. CONCLUSIONS

The main objective of this paper was to investigate a problem of contract preparation in many-to-many e-services. The process of contract composition and preparation is based on a combination of temporal and deontic logic. This paper argues that proper support during contract preparation is crucial for the process of forming short or long-term coalitions (i.e. virtual enterprises) to pursue market opportunities. This is especially important for small business as they are not required to have very sophisticated technical platforms to be able to participate in service provisioning.

Our future work in this area will also include monitoring of complex e-service execution, dynamic composition of services during run-time and analysis of the

accumulated experience on service execution. We envisage that the work presented in this paper, sets a good foundation for all of these future research challenges.

ACKNOWLEDGMENT

The work reported in this paper has been funded in part by the Cooperative Research Centres Program through the Department of the Prime Minister and Cabinet of the Commonwealth Government of Australia.

REFERENCES:

- Allen, J.F. (1981), "An interval based representation of temporal knowledge", in *Proc. 7th Int. Joint Conf. On Artificial Intelligence*, 221 – 226.
- Cole, J. et al. (2001), "Author Obligated to Submit Paper before 4 July: Policies in an Enterprise Specification", *Policy2001 workshop*, Bristol, UK, January.
- Dechter, R., Meiri, I., Pearl, J. (1991), "Temporal constraint networks", *Artificial Intelligence*, 49, 61-95.
- Durante, A. et al. (2000), "A Model for the E-Service Marketplace", *Hewlett-Packard Company*.
- Griffel, F. et al. (1998), "Electronic Contracting with COSMOS – How to Establish, Negotiate and Execute Electronic Contracts on the Internet", *EDOC'98 Workshop*, La Jolla, California, USA.
- Herring, C. and Milosevic, Z. (2001), "Implementing B2B Contracts Using BizTalk", *Proc. of HICSS-34 Conference*, Hawaii, Honolulu.
- ISO/IEC WD 15414. (1998), *Open Distributed Processing – Reference Model – Enterprise Viewpoint*, January.
- Jajodia, E.S. and Sripada, S. (Eds.) (1998), *Temporal Databases – Research and Practice, Lecture Notes in Computer Science*, 1399.
- Kuno, H. (2000), "Surveying the E-Services Technical Landscape", *Hewlett-Packard Company*.
- Lee, R.M. (1998), "A logic model for electronic contracting", *Decision Support Systems*, 4, 27-44.
- Mana, Z. and Pnueli, A. (1979) "The modal logic of programs", *Lecture Notes in Computer Science*, Vol.71, Springer Verlag, 385 – 411.
- McDermott (1982), "A temporal logic for reasoning about actions and plans", *Cognitive Science* 6, 101 - 155.
- Milosevic, Z. and Bond, a. (1995), "Electronic commerce on the Internet: What is still missing?", *Proc. of the 5th Conference of the Internet Society*, pg. 245-254, Honolulu.
- Nayak, N. et. al. (2001), "Virtual Enterprises – Building Blocks for Dynamic e-Business", *Workshop on Information Technology for Virtual Enterprises (ITVE)*, Gold Coast.
- von Wright, C.G. (1968), "an Essay in Deontic Logic and the General Theory of Action", *Acta Philosophica Fennica*, 21, North-Holland.