Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Theoretical Computer Science
ELSEVIER Electronic Notes in Theoretical Computer Science 130 (2005) 211–233
www.elsevier.com/locate/entcs

Early Verification of Computer Systems Temporal Properties

Paulo Sérgio Muniz Silva[1]

Departamento de Engenharia de Computação e Sistemas Digitais
Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, trav. 3, n.158 05508-900 São Paulo SP Brazil

Abstract

In early moments of computer systems development, computer engineers typically draw interaction diagrams, occasionally annotated with timing constraints, to reason about the specification of the system behavior. One of the most popular of these diagrams is the Message Sequence Chart (MSC). However, not always does the intended behavior described by MSCs correspond to their actual behavior. To help the formal verification of their actual behavior, i.e. their temporal properties, this paper describes an interpretation of basic timed MSCs in a temporal framework that formally represents, in a unified model, both the qualitative and the metric temporal information conveyed in these intuitive diagrams. The framework solves the verification problems in polynomial time and lays the foundation of an automatic tool.

Keywords: System behavior verification, message sequence chart, temporal reasoning.

1 Introduction

Typically, in early moments of computer systems development, computer engineers draw diagrams to reason about the behavior of the computer system that they will build. For example, when working with use case models, from the requirements viewpoint, software engineers usually draw sequence diagrams [27] to depict the realization of use cases flows of events, which show interactions between domain objects. The sequence diagram is based on the basic Message Sequence Chart (MSC) artifact, which is a well-known visual tool largely used to model the behavior of systems, representing sequences of

[1] Email: paulo.muniz@poli.usp.br

doi:10.1016/j.entcs.2005.03.012

events exchanged between system abstractions and formally defined in [13] [2]. With the exception of real-time computer engineers, the majority of computer engineers do not accomplish the verification of the temporal properties of these diagrams, even if the diagrams express the intended behavior that will guide the rationale behind important design decisions that they will make. An important fact contributing to this situation is that traditional verification and validation tools are not of much help when faced with the partial information available at these early stages [11]. How to rigorously verify the temporal properties of these intuitive diagrams?

MSCs are scenario-based specifications, i.e. descriptions of how system components, users and the environment work concurrently and interact in order to provide system functionality, presenting examples of the system expected behavior and main exceptions [29]. Although scenario specifications result in partial descriptions of system behavior, the current widespread practice uses them as behavior specifications. In the case of MSCs, the intended behavior is expressed as sequences of interactions. There are two fundamental approaches to scenario specification semantics: scenario descriptions as design documents and scenario descriptions as specifications [29]. This latter approach is more adequate in the early phase of requirements, introducing the problem of finding an adequate design for the specification and proving that the design satisfies the specification requirements [2]. We are interested in the scenario descriptions as specifications approach and in a particular subset of basic MSC properties that allows the usual expression of the intuitive temporal ordering of messages - a qualitative viewpoint - and exhibits simple timing constraints [3] [13] on message flows - a quantitative viewpoint. We propose a formal semantics of MSC that allows the verification of conflicts between the intended and the actual behavior of a scenario in a unified temporal perspective, which combines the qualitative and the quantitative temporal viewpoints. This integrated semantics of MSCs is the main contribution of the paper.

The qualitative temporal perspective stems from the fact that when intuitively stating and describing the behavior of the problem domain events, we usually say: "X sends the message $M2$ to Y **after** receiving the message $M1$ from Z", instead of immediately assigning particular time values to the sending and receiving events. Also important is that these kinds of events last a certain time, allowing us to say: "X sends the message $M4$ to W **during** the reception of the message $M3$ from Z". First, and intuitively, we abstract the system behavior into this qualitative framework, and only after that we try to assign quantitative (metric) time information, hopefully without violating

[2] The ITU standard also defines a High-Level Message Sequence Chart (HMSC), composed by basic MSCs.

the qualitative temporal statements. An appropriate temporal ontology that directly supports these qualitative temporal notions is Allen's time interval theory [1]. Allen's theory was developed in the context of the so-called 'Naive Physics' with its common-sense representations, by taking the notion of the interval of time as a primitive one. It has been popular in natural language understanding, planning, knowledge representation, and in other fields of AI research. Allen's theory is an algebra of binary relations on intervals, carrying qualitative temporal information and allowing a formal reasoning about such information. The qualitative interpretation of MSCs is defined within such a time framework, in which qualitative temporal constraints specify the relative position of paired events. On the other hand, the quantitative interpretation of MSCs places metric temporal bounds between paired events, expressing quantitative durations of time. Our interpretation is based on an integrated model having Allen's theory and a well-founded temporal metric theory as its underlying semantics [15], capturing their intertwined nature, i.e. their mutual influences, contrary to the majority of semantic models of MSCs that separate the semantics of MSCs without timing assignments from the semantics with timing assignments.

The proposed interpretation of MSCs supports the generation of consequences of both their qualitative and quantitative temporal properties, in such a manner that we can formally check the potential conflicts between the intended and the actual behavior specified in the charts. If the verification task is performed early in the development process (for example, from the requirements or analysis views), it helps computer engineers to uncover important specification faults before considering design decisions. Lastly and importantly, the verification task is done in polynomial time and the presented formal model lays the foundation of a verification tool.

Related Work. MSCs have been extensively studied in the last years. Current trends in the analysis of MSCs take various approaches: process algebra [20] and varieties of model checking [28] [2] [4] [3] [5], to mention just a few. Studies working with partially ordered event structures derived from MSCs [28] [24] are of special interest. However, these partially ordered structures do not handle metric information. Metric information is specially studied in [5] [3]. [3] defines an extension of basic MSCs to specify timing constraints, but the analysis of *race conditions* in MSCs, i.e. violations on the intended ordering of the events in MSCs due to their actual semantics - a qualitative analysis - is realized independently from the analysis of timing conflicts. This model originated versions of tool analyzers culminating in the UBET tool [18], which implements a rich set of features of basic MSCs complying with the ITU specification [12]. Unfortunately, UBET, and other popular tools that

followed (e.g. MESA [6], LTSA-MSC [19], EventStudio [10], among others), though presenting sophisticated features for the analysis of both MSCs and High-Level MSCs, do not carry on an actual integrated analysis of the race conditions and timing analysis. [3]

A preliminary version of the ideas underlying the present proposal appeared in [22]. In [23], we completely reformulated this preliminary approach, simplifying the interpretation and, most important, proposing a tractable solution for the verification problem that overcomes the complexity limitations of the former model. However, both models were pinned on the exclusive qualitative viewpoint, having Allen's theory as their foundation, but they paved the way to posit the richer combined qualitative and quantitative interpretation of temporal properties conveyed in MSCs. In section 2, we present the appropriate formal definition for the basic MSC. Sections 3, 4 and 5 present the qualitative, the metric and the combined interpretations of the defined MSC, respectively. Section 6 describes two simple experiments that show the application of the integrated model, and section 7 presents some conclusions and our current related research.

2 The Definition of MSCs

Firstly, we will present a brief overview of MSCs, quoting [3]. MSCs are a graphical representation which shows message exchanges between concurrent process abstractions within a system. Figure 1(a) shows a *basic* MSC extended with simple metric timing constraints on exchanged messages, the MSC type we will use as a model for the intended computer system partial behavior. Each vertical line has a start and an end symbol, and represents processes or autonomous agents ($P1, P2$ and $P3$). Each horizontal arrow describes a message sent from one process to another (a, b and c). The tail of an arrow corresponds to the event of sending a message, whereas the head corresponds to its receipt. Communication is one-to-one and asynchronous, and control flows independently within each process from the start symbol to the end symbol. In each process, the events are temporally ordered from top to bottom. The system terminates when all processes have terminated.

The behavior of an MSC is the set of sequences of sent and received messages, i.e., MSCs represent the intended behavior by the order of the exchanged messages between processes. The intended order does not necessarily represent the actual semantics of the MSC. Conflicts are likely to happen. For example,

[3] In fact, timing analysis features were excluded from the later more sophisticated versions of UBET. We did not have access to expensive commercial tools in order to analyze their capabilities.

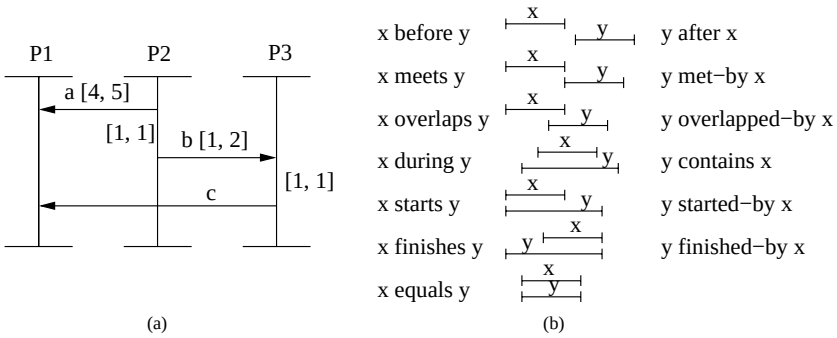

Fig. 1. A basic timed MSC and Allen's basic interval relations

if we ignore for the moment the metric timing constraints, it is not hard to see that there is a scenario for the MSC of Figure 1(a) in which message c arrives earlier than message a at process $P1$, conflicting with the intended order.

In the MSC of Figure 1(a) the label [4, 5] on message a specifies the lower and the upper bounds on the duration of message exchange. The label [1, 1] between the sending of message a and the sending of message b specifies the duration between these events, modeling an assumption about the computation on process $P2$. We call this extended MSC by *basic timed MSC*, as of [12].

On the other hand, qualitative temporal intuitions can be modeled by Allen's temporal structure [1], that captures two aspects of particular interest: the strict relative temporal knowledge (e.g. "X happens before Y", "X happens during Y", etc.) and the uncertainties of the information about the relationship between two events in time. The temporal structure is a simple and linear model of time. The original theory has the time interval as a primitive. Five axioms of the temporal structure and a complete set of thirteen intuitive binary relations between intervals - the Allen's basic relations - are defined. Figure 1(b) depicts these relations.

We will define a formal structure, the *basic time labeled MSC*, which captures the essential properties of the basic timed MSC. We extend [23] and [2] [4] appropriately, for the definitions of the timed properties of the MSC and other details.

Definition 2.1 Let $P = \{P_1, \ldots, P_n\}$ be the set of *processes*, and M be the set of *messages*. Let the label $!(i, j, a)$ denote the event "process P_i sends the message a to process P_j". Let the label $?(i, j, a)$ denote the event "process P_j receives the message a from process P_i". Let the label $\#(i)$ denote

[4] In [23] we only considered the non-timed properties of the MSC and based the definitions on [2].

the event "process P_i terminates by arriving at its bottom". Let the labels $[i,j], [i,j), (i,j]$,and (i,j), where $i,j \in \{0,1,\ldots,\infty\}$, denote the lower and upper bounds on the metric time interval between pairs of MSC events that may be closed, half-closed or open, respectively. Define the set $L_S = \{!(i,j,a) \mid i,j \in \{1,\ldots,n\} \ \wedge a \in M\}$ of *send labels*, the set $L_C = \{?(i,j,a) \mid i,j \in \{1,\ldots,n\} \wedge a \in M\}$ of *receive labels*, the set $L_B = \{\#(i) \mid i \in \{1,\ldots,n\}\}$ of *bottom labels*, the set $L = L_S \cup L_C \cup L_B$ as the set of *event labels*, the set L_T of *metric time labels*, and the set L_N of *next process event labels* (defined below).

A basic time labeled MSC over processes P is defined by:

- A set E of events partitioned into a set S of sending events, a set C of receiving events, and a set B of bottom events.
- A mapping $p : E \mapsto \{1,\ldots,n\}$ that maps each event to a process on which it occurs.
- A bijective mapping $f : S \mapsto C$ between sending and receiving events, matching each sending event with its corresponding receiving event.
- A bijective mapping $ne : E \mapsto E$ that maps each event on a process to its consecutive event on the same process. Each process event is connected to a unique consecutive event in the same process. This mapping is called *next process event*.
- A mapping $l : E \mapsto L$ which labels each event such that $l(S) \subseteq L_S$, $l(C) \subseteq L_C$, and $l(B) \subseteq L_B$. For consistency of labels, for all $s \in S$, if $l(s) = !(i,j,a)$ then $p(s) = i$ and $l(f(s)) = ?(i,j,a)$ and $p(f(s)) = j$. Also, for all $b \in B$, if $l(b) = \#(i)$ then $p(b) = i$.
- A mapping $h : E \times E \mapsto L_N$, which labels each next process events on each process P_i.
- A mapping $t : E \times E \longmapsto L_T$, which labels the mappings f and ne, and denotes metric time constraints between events. Note that this mapping is optional.

For each $i \in \{1,\ldots,n\}$, there is a total order $\preceq_i$ on the events of process P_i, that is, on the elements of $p^{-1}(i)$, such that the transitive closure of the relation $\preceq \doteq \cup_{i\in\{1,\ldots,n\}} \preceq_i \cup\{(s,f(s)) \mid s \in S\}$ is a partial order on E. This partial order is called *visual order*.

The total order $\preceq_i$ denotes the temporal order of the events of process P_i. The partial order $\preceq$ denotes the *visual order* of the MSC, enforcing the notion that "messages cannot travel back in time", and expresses the intended temporal behavior of the MSC.

The partial order corresponding to the MSC of Figure 1 is depicted in

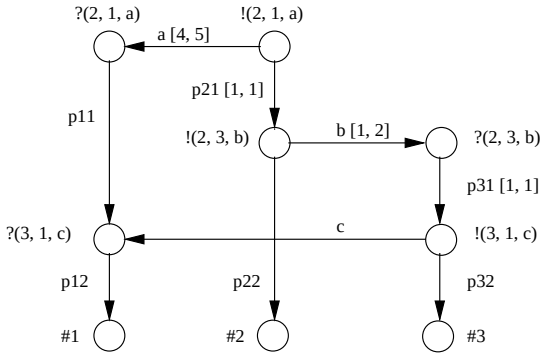

Fig. 2. The partial order of the MSC of Figure 1

Figure 2, where the nodes are sending, receiving, and bottom events, and the edges are messages and next process events.

The proposed interpretation of the basic time labeled MSC is realized in a classical model of time, where time is linear and time points are interpreted in a real line. A *time interval* X is represented as a pair of time points (x^-, x^+), such that $x^- < x^+$. An interval interpretation *I-interpretation* is the mapping of time intervals to pairs of distinct real numbers such that the beginning of an interval is strictly before the end of the interval [25]. In a basic time labeled MSC we have two fundamental intervals: the *message interval* and the *process interval*. The former is delimited by a sending event and a receiving event of a particular message. The latter is delimited by two consecutive events in a process. They are naturally defined in terms of the I-interpretation, i.e. by their endpoints. Two intervals stand in a particular *qualitative temporal relationship* defined in Allen's framework [1]. The difference of two time points expresses the distance in the time line (duration) between them, defining a *metric temporal relationship* that may be bounded from above and below.

Definition 2.2 In a basic time labeled MSC we have two types of time intervals: the *message interval* and the *process interval*.

- Define the mapping $u : E \mapsto \Re$, where $\Re$ is the set of real numbers. The message interval is the tuple (a, b), such that $a < b$, where $a, b \in \Re$, and for an $s \in S$, if $u(s) = a$ then $u(f(s)) = b$.
- Define the mapping $v : E \mapsto \Re$. The process interval is the tuple (a, b), such that $a < b$, where $a, b \in \Re$, and for $e_1, e_2 \in E$ and $(e_1, e_2) \in ne$, if $v(e_1) = a$ then $v(e_2) = b$.

The visual order, which expresses the intended temporal behavior of a basic MSC, relates messages with each other through processes. Messages are sent and received by processes. Therefore, the relative occurrence of messages

in the time dimension is determined by the order in which they are performed in processes. The qualitative and metric temporal relationships between then can each express certain *binary constraint satisfaction problems* (CSP) [21] or *networks of binary constraints*.

Definition 2.3 A network of binary constraints [21] is defined as a set X of n variables $\{x_1, x_2, \ldots, x_n\}$, a domain D_i of possible values for each variable, and binary constraints between variables. A binary constraint C_{ij}, between variables x_i and x_j, is a subset $C_{ij} \subseteq D_i \times D_j$. For networks built on Allen's framework, it is required that $(x_j, x_i) \in C_{ji} \iff (x_i, x_j) \in C_{ij}$. An *instantiation* of the variables in X is an n-tuple $(X_1, X_2, \ldots, X_n)$, representing the assignment of $X_i \in D_i$ to x_i. A *consistent instantiation* of a network is an instantiation of the variables such that the constraints between variables are satisfied. A network is *inconsistent* if no consistent instantiation exists.

The qualitative temporal relationships constitute a constraint network where each variable represents a temporal interval and the constraints represent the qualitative temporal knowledge in terms of Allen's framework [1]. The metric temporal relationships constitute a constraint network where each variable represents a time point and the constraints represent the metric temporal knowledge in terms of the temporal distance that may be bounded from above and below. The visual order of a basic time labeled MSC can be translated into these constraints networks that can be combined and solved, providing the actual temporal behavior of the MSC. The next sections present the qualitative and the metric temporal interpretations in terms of constraint networks, and the combined qualitative and metric interpretation.

3 The Qualitative Temporal Interpretation

The qualitative temporal interpretation represents the temporal knowledge in terms of Allen's framework [1].

Definition 3.1 Let I be the set of all mutually exclusive basic relations $\{b, bi, m, mi, o, oi, s, si, d, di, f, fi, eq\}$, where b stands for before, bi for after, m for meets, mi for met-by, o for overlaps, oi for overlapped-by, s for starts, si for started-by, d for during, di for contains, f for finishes, fi for finished-by, and eq for equals. The relation between two time intervals is any subset of I, representing a disjunction of the basic relations. The disjunction of all basic relations is denoted by $\top$ and the empty relation is denoted by $\bot$.

Allen's framework constitutes an algebra: the Allen's interval algebra. The algebra is based on the notion of relations between pairs of intervals.

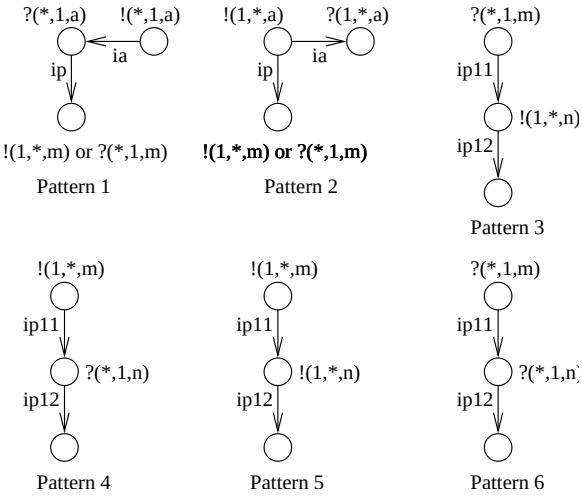

Fig. 3. Basic patterns of a time labeled MSC

Under the I-interpretation, we can express the basic relations in terms of endpoint relations. For example, the relation $X\, overlaps\, Y$ is equivalent to $x^- < y^-$, $x^- < y^+$, $x^+ > y^-$, $x^+ < y^+$. Allen's Interval Algebra **IA** is the algebra with underlying set 2^I (the power set of I), unary operator converse, binary operator intersection and binary operator composition. The intersection can be expressed as the set-theoretic intersection of the sets of basic relations. The composition is the union of the component-wise composition of the basic relations. Allen provides a composition table for the basic relations [1]. We can write $\{d, o, s\}$ to denote the disjunction of basic relations d, o and s. Therefore, $\{d, o\} \subset \{d, o, s\}$. Also, if X and Y are intervals and $X\{d, o, s\}Y$, then $Y\{di, oi, si\}X$.

In the visual order of a time labeled MSC, we can define the following fundamental patterns of qualitative relationships between a message interval and a process interval, depicted in Figure 3. In the figure, the symbol '$*$' denotes any process, the symbol '1' denotes the process of reference ("this process"), the symbol 'ia' denotes the label of the message (message interval), and the symbol 'ip' denotes the label of the next process event in the process of reference (process interval) [5].

Definition 3.2 In the visual order of every basic labeled MSC there are two patterns of qualitative relationships between message and process intervals, and four patterns of qualitative relationships between process intervals, de-

[5] In this paper, we simplify the patterns with respect to the approach taken in [23]. This is possible due to the introduction of bottom events in the time labeled MSC.

picted in Figure 3. A fundamental reasonable assumption is that a sending event is a controlled event in a process, which is only issued when a preceding event has occurred [3]. As a consequence, the visual order is not supposed to be guaranteed between pairs of receiving events, since they are not controlled events in a process, i.e. they may happen in any order.

The essence of the behavior representation in an MSC are messages, therefore we are primarily interested in the relative positions of messages intervals with respect to process intervals [23]. We proceed with the interpretation as follows: we fix the position of the process interval and vary the relative position of the message interval with respect to the process interval. The interpretation of the relationship between two process intervals is straightforward. Obviously, the mutual positions shall respect the patterns' configurations. The comparison of the resulting mutual positions between the intervals with Allen's basic interval relations of Figure 1b gives the interval relations. Let ia be the message interval delimited by the sending and the receiving events of message a, and let ip be the process interval delimited by events of the next process event p. It is easy to see that:

- Pattern 1. The message interval ia *meets* the process interval ip, i.e., $ia \{m\} ip$.
- Pattern 2. The message interval ia *starts*, or *equals*, or *is-started-by* the process interval ip, i.e., $ia \{s, si, eq\} ip$.
- Patterns 3, 4, and 5. The process interval $ip11$ *meets* the process interval $ip12$, i.e., $ip11 \{m\} ip12$.
- Pattern 6. The relationship between process intervals $ip11$ and $ip12$ present unknown temporal information, i.e. they may convey any interval relation, or $ip11 \top ip12$.

The set of message and process intervals from a labeled MSC, and the binary relations between each pair of them, constitute a binary constraint network where the nodes are messages or process intervals and the edges are the interval relations corresponding to the basic pattern with which the message and process intervals match. We call this network the *Qualitative Interval Calculus Network (QICN)*. Figure 4 illustrates the derivation of a QICN from the time labeled MSC of Figure 2, where ia, ib and ic, denote the message intervals to the messages a, b and c, respectively; and $ip11$, $ip12$, $ip21$, $ip22$, $ip31$ and $ip32$ denote the process intervals to the next process events $p11$, $p12$, $p21$, $p22$, $p31$ and $p32$ of processes $P1$, $P2$ and $P3$, respectively. Walking through all pairs of relations in the visual order of the MSC, we derive the following interval relations in the resulting QICN: $ia \{m\} ip11$ (pattern 1), $ia \{s, si, eq\} ip21$ (pattern 2), $ip21 \{m\} ip22$ (pattern 5), $ib \{s, si, eq\} ip22$ (pat-

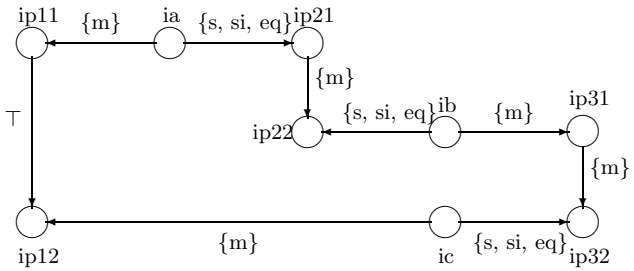

Fig. 4. The resulting QICN

tern 2), $ib\,\{m\}\,ip31$ (pattern 1), $ip31\,\{m\}\,ip32$ (pattern 3), $ic\,\{s, si, eq\}\,ip32$ (pattern 2), $ic\,\{m\}\,ip12$ (pattern 1), and $ip11\,\top\,ip12$ (pattern 6).

Let us define the *Interval Algebra Network* (**IA** *network*) [31]. An **IA** *network* is a network of binary constraints where the variables represent time intervals and the binary constraints between variables are represented implicitly by disjunctions of the basic relations.

Proposition 3.3 *The QICN derived from a time labeled MSC is an* ***IA*** *network.*

Proof. Let M and N denote the sets of message intervals and process intervals, respectively. Definition 2.2 states that their elements are time intervals. The QICN is a set of variables $\{x_1, x_2, \ldots, x_n\}$ with possible values taken from the domain $D = M \cup N$, that is, the values of the x_i are time intervals. The relations between the QICN variables are disjunctions of basic relations, since the derivation process of the QICN relations is carried out in accordance with the basic patterns, delivering interval relations. Consequently, the QICN is an **IA** network. □

The fundamental reasoning problems in an **IA** network are [30]: find a scenario that is consistent with the information provided, and find the feasible relations between all pairs of intervals. We are interested in finding the feasible relations between all pairs of intervals mostly, that is, in finding the deductive consequences of the qualitative temporal knowledge represented in an MSC.

Definition 3.4 A basic relation $r \in W$ is feasible with respect to a network W if and only if there exists a consistent instantiation of the network where r is satisfied [30]. Given an **IA** network W, the set of feasible relations between two variables x_i and x_j in the network is the set consisting of all and only the $r \in W$ that are feasible. The minimal network representation w, of a network W, is the network for which w_{ij} is the set of feasible relations between variables x_i and x_j in W, for every $i, j = 1, \ldots, n$. Determining the feasible relations in

W can be viewed as determining the deductive consequences of the qualitative temporal knowledge.

Constraint satisfaction techniques are used to solve the reasoning problem. For the **IA** network, finding a consistent scenario and finding the feasible relations are NP-complete problems and intractable in the worst case [30] [31]. In fact, Allen's algorithm [1] is an approximation solution for the network in the full algebra (**IA** Algebra). Subsequent research has focused on designing more efficient algorithms or identifying tractable special cases of the full algebra for which there are exact solutions [9] [16] [31], to mention just a few. The full **IA** algebra contains $2^{13} = 8192$ possible relations between intervals. Subclasses of the **IA** algebra are obtained by considering their subsets giving 2^{8192} subclasses. In many applications the full algebra is not necessary and restricted classes of interval algebras have an exact solution to the temporal network [6] .

In [23] we defined the subalgebra $\mathbf{SA_{ICN}}$ that meets all the requirements for the QICN binary constraints. We proved that $\mathbf{SA_{ICN}} \subset \mathbf{SA_C} \subset \mathbf{IA}$, where $\mathbf{SA_C}$ is the *Continuous Pointizable Interval Algebra* [31], and also proved that the $\mathbf{SA_{ICN}}$ is a tractable special case of the **IA** algebra. Hence, the finding of the feasible relations between all pairs of intervals of the QICN is exact and solved by any path consistency algorithm (for example, [16]), which usually requires polynomial $O(n^2)$ time (n is the number of intervals).

The *Qualitative Interval Calculus Network (QICN)* is formally defined as follows:

Definition 3.5 Let $\mathbf{IA_{QICN}} \subset \mathbf{IA}$ be the subset of relations $\{\{m\}, \{s, si, eq\}, \top\}$ which are allowed to occur in the translation of a time labeled MSC into a QICN. Let the subalgebra $\mathbf{SA_{QICN}}$ be the least subalgebra of **IA** containing $\mathbf{IA_{QICN}}$, which is closed under operators converse, intersection and composition. Since $\mathbf{SA_{ICN}}$ is the least subalgebra of **IA** containing $\mathbf{IA_{ICN}} = \{\{m\}, \{mi\}, \{s, si, eq\}, \{f, fi, eq\}\top\}$ [23], and $\mathbf{IA_{QICN}} \subset \mathbf{IA_{ICN}}$, $\mathbf{SA_{QICN}}$ is also a tractable special case of the **IA** algebra.

Let V be the set of intervals. Let L_V be the set of interval labels. Let R be the set of the interval relations of $\mathbf{IA_{QICN}}$. Let L_R be the set of QICN interval relation labels, whose members denote the interval relations of $\mathbf{IA_{QICN}}$. The QICN is a binary constraint network defined by:

- A set M of *message intervals* and a set N of *next process event intervals* that partition the set V.
- A bijective mapping $h : M \mapsto N$ between message and next process event

[6] The approach taken in [22] was intractable, since we worked in the full algebra.

intervals.

- A mapping $z : V \mapsto L_V$ that labels each interval.
- A mapping $r : V \times V \mapsto L_R$ that labels each relation between intervals. r is called an *interval relation*.
- If there is an interval relation between two nodes with a particular set of relations, its *converse interval relation* is the set whose elements are the converse of each element from the former set.

The appendix depicts the underlying set of the subalgebra $\mathbf{SA_{QICN}}$, containing all possible relations that may occur in the solutions of QICN networks.

4 The Metric Temporal Interpretation

The metric temporal interpretation represents the temporal knowledge in terms of the temporal distance between time points that may be bounded from above and below. It constitutes a metric temporal network and, consequently, a metric temporal constraint satisfaction problem defined as follows [8]:

Definition 4.1 A network of binary metric temporal constraints consists of a set of variables $\{x_1, x_2, \ldots, x_n\}$ and a set of *unary* and *binary constraints*. A unary constraint T_i restricts the domain of variable x_i to the given set of metric time intervals $\{I_1, \ldots, I_k\} = \{[a_1, b_1]\}, \ldots, [a_k, b_k]\}$, representing the disjunction $(a_1 \leq x_i \leq b_1) \vee \ldots \vee (a_k \leq x_i \leq b_k)$. A binary constraint T_{ij} constrains the permissible values for the distance $x_j - x_i$, representing the disjunction $(a_1 \leq x_j - x_i \leq b_1) \vee \ldots \vee (a_k \leq x_j - x_i \leq b_k)$. The network can be represented by a *directed constraint graph*, where nodes represent variables and an edge $i \rightarrow j$ indicates that the constraint T_{ij} is specified between x_i and x_j; it is labeled by the interval set. Each input constraint T_{ij} implies an equivalent constraint T_{ji}. We may treat each unary constraint T_i as a binary constraint T_{0i} relative to a special time point x_0 representing the "beginning of the time", assuming $x_0 = 0$ for simplicity.

A tuple $x = (a_1, \ldots, a_n)$ is called a *solution* if the assignment $(x_1 = a_1, \ldots, x_n = a_n)$ does not violate any constraint. A value v is a feasible value for a variable x_i if there exists a solution in which $x_i = v$. The set of all feasible values of a variable is called the *minimal domain*. A minimal constraint T_{ij} between x_i and x_j is the set of all feasible values for $x_j - x_i$. A network is called the *minimal network* if and only if its domains and constraints are minimal.

The first problem is to determine the consistency of a metric constraint

network. If it is consistent, the interesting problems to be solved are [8]:

- What are the possible times at which x_i could occur? In other words, what is the minimal domain of x_i?
- What are all the possible relationships between x_i and x_j? In other words, what is the minimal constraint between x_i and x_j?

It is known that the solutions for these problems are NP-hard [8]. However, the basic time labeled MSC does not represent metric temporal constraints as disjunctions, but single metric intervals between events. A network of binary metric temporal constraints in which all the constraints specify a single metric interval is called a *simple temporal problem* (STP) in the literature.

Definition 4.2 A simple temporal problem [8] is a network of binary metric temporal constraints in which each edge $i \rightarrow j$ is labeled by a single metric interval $[a_{ij}, b_{ij}]$ that represents the constraint $a_{ij} \leq x_j - x_i \leq b_{ij}$. Alternatively, the constraint can be expressed as the pair: $x_j - x_i \leq b_{ij}$, and $x_i - x_j \leq -a_{ij}$. An STP can be associated with a directed edge-weighted graph $G_d = (V, E_d)$, called a *distance graph*, where each edge $i \rightarrow j \in E_d$ is labeled by a weight a_{ij} representing the linear inequality $x_j - x_i \leq a_{ij}$.

An important theorem is that an STP T is consistent if and only if its distance graph G_d has no negative cycles [17]. Another important result is the theorem:

Theorem 4.3 *Let G_d be the distance graph of a consistent STP, T. The equivalent STP, M, defined by $\forall i, j M_{ij} = \{[-d_{ji}, d_{ij}]\}$, is the minimal network representation of T, and the set of feasible values for variable X_i is $[d_{i0}, d_{0i}]$. This new network M is called the* d-graph *of G_d, where each edge $i \rightarrow j$ is labeled by the shortest-path length d_{ij} in G_d. The d-graph corresponds to a more explicit representation of the STP.*

Proof. See [8]. □

The d-graph of an STP can be constructed by applying Floyd-Warshall's ALL-PAIRS-SHORTEST-PATHS algorithm [7] to the distance-graph of an STP [8]. Since the algorithm has time $O(n^3)$, where n is the number of variables, and detects negative cycles, it constitutes a polynomial time algorithm for determining the consistency of an STP, and for computing both the minimal domains and the minimal network, according to theorem 4.3. The assembling of a solution, i.e. the assembling of the minimal domains and the minimal network, requires only $O(n^2)$ time [8]. Therefore, finding a solution runs on $O(n^3)$ time.

The visual order graph of a basic time labeled MSC with only their metric time labels is abstracted into a metric constraint graph in which the variables are the events of the corresponding MSC. Metric temporal relationships are denoted by the metric time labels. We define the *Metric Interval Calculus Network (MICN)* that allows the reasoning about the metric temporal information conveyed in a basic time labeled MSC.

Definition 4.4 The *Metric Interval Calculus Network (MICN)* is a directed edge-weighted graph $MICN = (V, E_m)$, which has the same node set and node labels of the visual order graph of the corresponding basic time labeled MSC[7], and edges $i \rightarrow j$ labeled by a weight w_{ij}, representing the linear inequalities[8] $X_j - X_i \leq w_{ij}$ or $X_j - X_i < w_{ij}$. Consequently, it is a distance graph. Note that if a metric constraint between two nodes is bounded from above and below, there are two edges connecting these nodes representing the constraint $a_{ij} \leq X_j - X_i \leq b_{ij}$, that can be expressed as the pair of inequalities: $X_j - X_i \leq b_{ij}$ and $X_i - X_j \leq -a_{ij}$.

Proposition 4.5 *The $MICN$ can be solved in polynomial time for consistency and the findings of minimal domains and the minimal network.*

Proof. Trivial. Since, by definition 4.4, the $MICN$ is a distance graph, we can derive a d-graph from it by applying Floyd-Warshall's ALL-PAIRS-SHORTEST-PATHS algorithm that runs in time $O(n^3)$. The d-graph allows the computing of minimal domains and the minimal network in time $O(n^2)$ [8]. Therefore, finding a solution requires $O(n^3)$ time. □

5 The Combined Interpretation

Our goal is to provide a general framework that combines both temporal interpretations. Kautz and Ladkin [15] showed how metric and qualitative (Allen-style) temporal constraint networks can be integrated into a constraint-based reasoning system. Let L_Q be the language that expresses qualitative temporal formulas and L_M be the one that expresses metric temporal formulas. They demonstrated that the language $L_Q \cup L_M$ constitutes a practical temporal language that expresses both temporal representations. The below-presented al-

[7] Typically, there is no practical interest in the assignment of metric time intervals between a sending or a receiving event and the bottom event that lie on the same process.

[8] In fact, [8] proved that the Floyd-Warshall's algorithm finds d-graphs for a network with inequalities of the form $X_j - X_i \leq w_{ij}$. As we will see in the next section, we adopt the Kautz and Ladkin's approach [15] that modifies the algorithm to also handle the strict inequality.

gorithm COMBINED-METRIC-ALLEN shows a constraint satisfaction algorithm for the union of the two languages. They proved the soundness of the algorithm and also proved that it terminates in $O(n^2(e+n^3))$ time, where n is the number of intervals that appear in both networks (QICN and MICN), and e is the time required to compute the QICN. Most important, they also proved that if the qualitative network (Allen's network) is pointizable the algorithm iterates in no more than two times. As showed above, we proved elsewhere [23] that the QICN is (continuous) pointizable. Therefore the solution of the temporal problems for the union of both networks inherits this important performance property. The method adopted by [15] is to separately solve each network, derive new Allen constraints from the metric network, add these constraints to the qualitative network, derive new metric constraints from the qualitative network, add these constraints to the metric network, and repeat this process until no new statements can be derived. The queries about temporal properties are answered by examining the adequate network. Kautz and Ladkin present the optimal translations between L_Q and L_M, and a complexity analysis of the combined inference algorithm. They proved that their algorithm METRIC-TO-ALLEN, that translates L_M to L_Q, is sound and runs in $O(n^3)$ time, where n is the number of intervals of the quantitative network. They also proved that their algorithm ALLEN-TO-METRIC, that translates L_Q to L_M, is sound and runs in $O(e+n^2)$ time, where e is the time needed to solve the qualitative network and n is the number of intervals of the qualitative network.[9] To handle strict inequalities, Floyd-Warshall's ALL-PAIRS-SHORTEST-PATHS algorithm is appropriately modified.

COMBINED-METRIC-ALLEN(M, A) =
input: simple metric network M and simple Allen network A
output: networks $M^{'}, A'$ implied by $M \cup A$
    repeat
        $A' :=$ METRIC-TO-ALLEN$(M) \cup A$
        $M^{'} :=$ ALLEN-TO-METRIC$(A') \cup M$
        $M := M'$; $A := A'$
    until $A = A'$ and $M = M'$
    return M', A'
end COMBINED-METRIC-ALLEN

[9] Due to space limitations, we will not reproduce both algorithms. They can be found in [15].

6 Simple experiments

We will present two simple working experiments. Kautz' Metric/Allen Time System - MATS [14], a Common Lisp program based on the theory described in [15], solves the reasoning problems of combined temporal constraint networks. The original code, written in an old version of Common Lisp, has been adequately updated to ANSI Common Lisp by the author of this paper and used to run all the examples described below. The appendix depicts the Experiment 1.2 code submitted to MATS.

Qualitative constraints are entered in MATS by the following Lisp form: `(asserta INTERVAL1 ALLEN-RELATION INTERVAL2)`, where `INTERVAL1` and `INTERVAL2` are time intervals constrained by an `ALLEN-RELATION` (a basic relation or a disjunction of basic relations). Metric constraints take the form of difference inequalities on point forms. A point form is the function `left` or `right` followed by the name of the interval, representing the pair of time points of an interval. A metric constraint is entered by the following form: `(assertm INEQUALITY-FORM)`. `INEQUALITY-FORM` may be one of the three forms of an inequality: the upper and lower bounds for a time point, a difference between two time points or a duration of an interval. The constraint propagation is realized by the function `reduce` and there is a rich set of functions to query the temporal data base, displaying information about the qualitative and metric networks.

Experiment 1: the MSC of Figure 1. First, we obtain the partial order of the MSC (depicted in Figure 2). After, we obtain the resulting QICN (figure 4) and the resulting MICN. An automatic tool based on the formal definitions provided can directly handle these transformations.

Experiment 1.1. Solving only the QICN, not taking the metric information into consideration. This is the typical problem solved by the above-mentioned tools and the focus of [22] and [23]. The solution shows the following qualitative relations between messages a, b and c: R_1 is ia $\{di, fi, b, m, o\}$ ib; R_2 is ia $\{di, fi, b, m, o\}$ ic; R_3 is ib $\{b\}$ ic. For the sake of simplicity, let *!message* denote a sending event of a message in a process and *?message* denote a receiving event of a message in a process. The visual order of the MSC requires that: $?a < ?c$, $!a < !b$ and $?b < !c$. Expressing the interval relations in terms of endpoint relations, the interval relations R_1 and R_3 guarantee $!a < !b$ and $?b < !c$, respectively. However, interval relation R_2 does not guarantee $?a < ?c$. In fact, the basic interval relation $di \in R_2$ says that $?c < ?a$, which contradicts the intended behavior expressed in the visual order.

Experiment 1.2. Solving the combined QICN and MICN and considering the metric information. The solution now shows the following qualitative

relations between messages a, b and c: R_1 is ia $\{di\}$ ib; R_2 is ia $\{di, fi, m, o\}$ ic; R_3 is ib $\{b\}$ ic. The metric inequalities show: $duration(c) > 0$, $3 \leq !c \leq 4$, and $3 < ?c < \infty$. A manual inspection, working with all the combination of the mutual interval positions in a time scale, demonstrates the correctness of the solution. The interval relation R_2 does not yet guarantee $?a < ?c$, because $di \in R_2$, but R_1 and R_2 have different values w.r.t. Experiment 1.1. This result demonstrates the effective mutual influence of both networks, and opens a richer scope for the scenario behavior analysis.

Experiment 1.3. Solving the combined QICN and MICN and considering the following modifications in the metric information of the MSC of Figure 1: a duration of message a with strictly 4 time units and a duration of message c with $[1, 2]$ time units as lower and upper bounds, respectively. The solution shows the following qualitative relations between messages a, b and c: R_1 is ia $\{di\}$ ib; R_2 is ia $\{fi, m, o\}$ ic; R_3 is ib $\{b\}$ ic. The metric inequalities show: $1 \leq duration(c) \leq 2$, $3 \leq !c \leq 4$, and $4 < ?c < 6$. Now, the interval relation R_2 guarantees $?a < ?c$, stating that these particular timing assignments do not contradict the intended behavior expressed in the visual order.

Experiment 2. Another example is taken from [29], modified to carry metric time information. Figure 7(a) depicts an MSC that specifies the general intended behavior for a simple system that controls employees entering a secure building. We assume that the Door, the Security System and the Camera are autonomous systems, and the communication between them is asynchronous. Figure 7(b) shows the correspondent partial order, without the event labels for the sake of space economy, and Figure 7(c) shows the resulting QICN.

Experiment 2.1. Solving only the QICN, not taking the metric information into consideration. The solution shows interesting relations: *istartRecording* $\{di, fi, b, m, o\}$ *iunlock*, and *istopRecording* $\{d, f, bi, mi, oi\}$ *istartRecording*. The interval relation *istartRecording* $\{di\}$ *iunlock* does not guarantee that the camera records when an employee enters the building (the reception of the *unlock* message occurs before the reception of the *startRecording* message). The interval relation *istopRecording* $\{d\}$ *istartRecording* says that the *Camera* stops recording before it starts recording. This may show a flaw in the specification.

Experiment 2.2. Solving the combined QICN and MICN and considering the metric information. Now, the solution states: *istartRecording* $\{di, fi, m, o\}$ *iunlock*, and *istopRecording* $\{bi\}$ *istartRecording*. The metric inequalities show: $-2 \leq ?istartRecording - ?iunlock \leq 1$, and $2 < ?istopRecording - ?istartRecording < \infty$. Though this behavior is more realistic because the camera stops the recording after it started recording (the duration of the

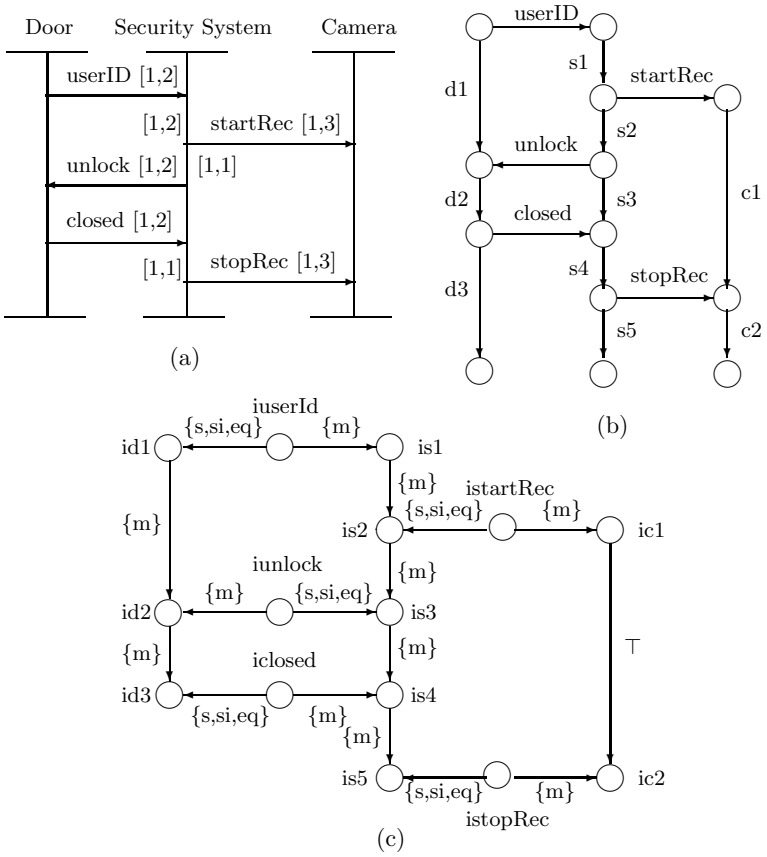

Fig. 5. A simple secure building control MSC, its visual order and QICN

recording is greater than 2 time units), it does not guarantee that the camera starts the recording when an employee enters the building (the reception of the *startRecording* message may occur 1 unit of time after the reception of the *unlock* message).

Experiment 2.3. Solving the combined QICN and MICN and considering the duration of process interval *id2* with $[10, 12]$ time units as lower and upper bounds, respectively. The qualitative relations have the same values as in Experiment 2.2. The metric inequalities show: $-2 \leq ?istartRecording - ?iunlock \leq 1$, and $12 \leq ?istopRecording - ?istartRecording \leq 20$. This behavior says that the duration of the recording is within the range $[12 \ldots 20]$ time units, but still does not guarantee that the camera starts the recording when an employee enters the building. The continuation of this result suggests a flaw in the specification.

7 Conclusions

In this paper, we have presented a framework to formally verify the behavior of computer systems partially described by MSCs in early moments of their development, where traditional verification and validation tools work under great difficulties. Our approach focused on the temporal properties of simple MSCs that carry timing assignments, providing an interpretation that integrates both qualitative and quantitative (metric) temporal information in a unique framework, contrary to the majority of approaches that separates both dimensions and their analyses, not considering their intertwined nature. The verification task is placed in the requirements viewpoint where important design decisions have not been made yet. Based on the verification results, computer engineers have an opportunity to catch specification faults early in the project and a rich set of actual behavior scenarios to be considered in the design decisions. We proved that the the verification task - both qualitative and metric - is exact and is carried out in polynomial time. The interpretation lays the foundation of an automated verification tool.

Currently, we are working on two goals: (1) an extension to the definition of the time labeled MSC to handle a richer set of requirements defined in [13]; (2) the managing of time labeled MSC composition (multiple MSCs) through High-Level MSCs [13], used to describe more complex behaviors.

Acknowledgments. Thanks to Henry Kautz for making the MATS code available [14] used in the experiments of this paper, and for allowing the changes in it.

References

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] R. Alur, K. Etessami, and M. Yannakakis. Inference of message sequence charts. In *22nd International Conference on Software Engineering*, page 304, 2000.

[3] R. Alur, G. J. Holzmann, and D. Peled. An analyzer for message sequence charts. In T. Margaria and B. Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, number 1055 in LNCS, pages 35–48. Springer-Verlag, 1996.

[4] R. Alur and M. Yannakakis. Model checking of message sequence charts. In *CONCUR '99: Concurrency Theory, Tenth International Conference*, number 1664 in LNCS, pages 114–129. Springer-Verlag, 1999.

[5] H. Ben-Abdallah and S. Leue. Expressing and analyzing timing constraints in message sequence chart specifications. Technical Report 97-04, Department of Electrical and Computer Engineering, University of Waterloo, 1997.

[6] H. Ben-Abdallah and S. Leue. Mesa: Support for scenario-based design of concurrent systems. In *TACAS-98*, page 111, 1998.

[7] T. H. Cormen et al. *Introduction to algorithms.* The MIT Press, USA, 2001.

[8] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.

[9] T. Drakengren and P. Jonsson. Eight maximal tractable subclasses of Allen's algebra with metric time. *Journal of Artificial Intelligence Research*, 7:25–45, 1997.

[10] EventHelix. *EventStudio 2.0 - User Guide.* EventHelix.com Inc., USA, 2004.

[11] G Holzmann. Early fault detection tools. In T. Margaria and B. Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, 1996.

[12] ITU-T. *Recommendation Z.120. Message sequence charts (MSC'96).* ITU-TS, Geneva, Switzerland, April 1996.

[13] ITU-T. *Recommendation Z.120. Message sequence charts (MSC'99).* ITU-TS, Geneva, Swiss, November 1999.

[14] H. Kautz. *Metric/Allen Time System - MATS.* ATT Bell Laboratories, Murray Hill, NJ, USA, 1991.

[15] H. Kautz and P.B. Ladkin. Integrating metric and qualitative temporal reasoning. In *Proceedings of AAAI-91*, pages 241–246, Anaheim, CA, USA, 1991.

[16] P. B. Ladkin and R. Maddux. On binary constraint problems. *Journal of the ACM*, 41(3):435–469, 1994.

[17] C. E. Leiserson and J. B. Saxe. A mixed-integer linear programming problem which is efficiently solvable. In *21st Anual Allerton Conference on Communication, Control and Computing*, page 204, 1983.

[18] Lucent. *Message Sequence Editor version 5.2 - User Guide.* Lucent Technologies - Bell Labs Innovation, USA, 1999.

[19] J. Magee and J. Kramer. *Concurrecy: State Models and Java Programs.* John Wiley and Sons Ltd., USA, 1999.

[20] S. Mauw and M. A. Reniers. Operational semantics for msc'96. *Computer Networks and ISDN Systems*, 31(17):1785–1799, 1999.

[21] U. Montanari. Networks of constraints: fundamental properties and applications to picture processing. *Information Science*, 7:95–132, 1974.

[22] P. S. Muniz Silva. Extended message sequence charts with time-interval semantics. In *Proceedings of the Fifth International Workshop on Temporal Representation and Reasoning*, pages 37–44, Sanibel Island, FL, USA, 1998.

[23] P. S. Muniz Silva. Early verification of software behavior in a time interval framework. In *Proceedings of the 17th Brazilian Symposium on Software Engineering*, pages 241–255, Manaus, AM, Brazil, 2003.

[24] A. Muscholl and D. Peled. Message sequence graphs and decision problems in mazurkiewicz traces. In *Proceedings of MFCS*, number 1672 in LNCS, pages 81–91. Springer-Verlag, 1999.

[25] B. Nebel and H-J. Bürckert. Reasoning about temporal relations: a maximal tractable subclass of Allen's interval algebra. *Journal of the ACM*, 42(1):43–66, 1995.

[26] B. Nebel and H-J. Bürckert. Software for machine assisted analysis of Allen's algebra, available from ftp.informatik.uni-freiburg/documents/papers/ki/, program allen-csp-solving.programs.tar.gz. 1995.

[27] OMG. *OMG Unified Modeling Language Specification - version 1.5.* Object Management Group, Inc., USA, September 2003.

[28] D. Peled. Specification and verification using message sequence charts. *Electronic Notes in Theoretical Computer Science*, 65(7), 2002.

[29] S. Uchitel. *Incremental Elaboration of Scenario-Based Specifications and Behaviour Models Using Implied Scenarios*. PhD thesis, Imperial College of Science, Technology and Medicine. University of London, Department of Computing, 2003.

[30] P. G. van Beek. Reasoning about qualitative temporal information. *Computational Intelligence*, 58:297–326, 1992.

[31] P. G. van Beek and R. Cohen. Exact and approximate reasoning about temporal relations. *Computational Intelligence*, 6(3):132–144, 1990.

Appendix

Follows a fragment of the code used to enter the QICN and the MICN constraint forms of Experiment 1.2 (Common Lisp code).

```
;; This section defines the QICN forms common to
;; Experiments 1.1, 1.2 and 1.3.
(asserta ia m ip11)
(asserta ia (s si =) ip21)
(asserta ip21 m ip22)
(asserta ib (s si =) ip22)
(asserta ib m ip31)
(asserta ip31 m ip32)
(asserta ic (s si =) ip32)
(asserta ic m ip12)
(asserta ip11 any ip12)     ; 'any' means all the basic relations

;; This section defines the MICN forms of Experiment 1.2

;; Time of reference.  t0 is the 'beginning of the time'.
(assertm 0 <= right t0 <= 0)
;; The time starts 'running' at the sending of message 'a'.
(asserta t0 m ia)

;; MICN forms
(assertm 4 <= duration ia <= 5)
(assertm 1 <= duration ip21 <= 1)
(assertm 1 <= duration ib <= 2)
(assertm 1 <= duration ip31 <= 1)
```

Follows the underlying set of subalgebra $\mathbf{SA_{QICN}}$ generated by the `aclose` tool [26], which contains all possible relations that occur in the binary constraints of solved QICN networks.

$\{\bot, \{eq\}, \{b\}, \{bi\}, \{d\}, \{di\}, \{o\}, \{oi\}, \{m\}, \{mi\}, \{s\}, \{si\}, \{f\}, \{fi\}, \{b, o, m\}, \{bi, oi, mi\}, \{d, o, s\}, \{di, oi, si\}, \{b, d, o, m, s\}, \{bi, di, oi, mi, si\}, \{eq, s, si\}, \{d, oi, f\}, \{di, o, fi\}, \{bi, d, oi, mi, f\}, \{b, di, o, m, fi\}, \{eq, f, fi\}, \{eq, d, di, o, oi, s, si, f, fi\}, \{eq, b, d, di, o, oi, m, s, si, f, fi\}, \{eq, bi, d, di, o, oi, mi, s, si, f, fi\}, \top\}$.