A Multi-agent Based Architecture to Qualitative Temporal Constraint Reasoning

TAHAR GUERRAM¹, RAMDANE MAAMRI² and ZAIDI SAHNOUN²

¹ Department of computer Science
University of Oum El Bouaghi, Algeria
Email: tahar_guerram@yahoo.fr

² LIRE Laboratory, Department of Computer Science
University Mentouri of Constantine, Algeria
Email: rmaamri@umc.edu, Email: sahnounz@yahoo.fr

Abstract: - Qualitative Reasoning (Q.R) is an inferring technique using a qualitative model to derive new qualitative knowledge. It’s closer to human reasoning and offers the advantage to coping with incomplete information and allows to predict the studied system’s behaviour. So, the main benefit of Q.R is that it is possible to provide an approximate solution to a given real world problem when all detailed precise information is unavailable or unnecessary. Qualitative temporal reasoning uses imprecise or vague temporal information and can be applied to qualitative task planning and scheduling. In this paper, we propose a multi agent based approach to qualitative temporal reasoning where available temporal information about actions is represented by a qualitative constraint network using Allen’s qualitative algebra.

Key Words: multi agent systems, Allen’s Algebra, qualitative constraint networks, qualitative temporal reasoning.

1 Introduction
Artificial intelligence aims to automate the reasoning as a principal component of human beings cognitive abilities what gave rise to knowledge based systems. To build such systems, we use a knowledge representation language and define an inference method to reason on represented knowledge. Time is a very imporing factor to take into account in real world applications modeled as dynamic systems in order to be able to highlight and predict their behaviors.

Many examples of temporal reasoning that require time representation and reasoning could be given as below [1]:
— Given a description of the world at a certain time t, plan a set of actions that can be executed in a certain order starting at t, to achieve a desired goal;
— Schedule a set of given activities to meet some constraints imposed on the order, duration, and temporal position or separation of the activities, such as a stipulated deadline;
— Given (possibly incomplete or uncertain) information about the temporal relations holding between events or facts in the represented domain, answer temporal queries about other entailed relations; for example, queries about the possibility or the necessity that two particular future events will temporally overlap or about the shortest temporal distance separating two events that have occurred. A temporal reasoning system should consist of a temporal knowledge base, a routine to check its consistency, a query-answering mechanism, and an inference mechanism capable of discovering new information [2]. When temporal information about domain variables is imprecise or uncertain, qualitative temporal reasoning is then applied. In our case, domain variables are
finite intervals between which Allen’s temporal relations are defined. This temporal information is firstly analyzed and then represented as a qualitative temporal constraint network (Q.T.C.N) which is a temporal graph when each node is labeled by an action and a time interval, and each arc is labeled by a temporal relation between two nodes. To reason on this qualitative model, we use constraint based problem solving techniques such as arc consistency, path consistency and backtracking. Applied to (Q.T.C.N), these techniques suffer of computational complexity which is NP Hard. In order to apprehend this computational problem, we propose to use multi agent’s paradigm which models system as a collection of interacting entities called agents using concepts like autonomy, communication, parallelism and cooperation. Our paper is organized as follows. Section 2 discusses related work done up to now in agent based qualitative temporal reasoning area. We devote section 3 to give highlights on qualitative temporal reasoning frameworks. Section 4, gives the description of our multi agent based approach to qualitative constraint reasoning. In section 5, we discuss implementation aspects of our approach. Finally, section 6, enables us to conclude our paper and to give some perspectives.

2 Related work

In the paper untitled “Multi agent oriented constraint satisfaction” [3], The authors proposed a multi agent system approach called ERA (abbreviation of Environment, Reactive rules and Agents) based on SWARM formalism for simulating distributed multi agent systems. Each agent of the system is reactive and plays the role of a variable and then moves in this variable’s domain. Agents interact in order to reach a global solution state. For each state, which corresponds to a consistent instantiation of the set of variables, the environment records a number called the violation constraint number for each value of the domains for all variables so that when perceived by the agents, these last could modify their behaviors (change their positions in the domains) in the objective to reach the “position zero” corresponding to a violation constraint number equal to zero. If it exists, a complete solution is obtained when all agents reach their zero positions, else an approximate solution is proposed. The ERA approach was tested on n-queen and coloring problems and the computational results were best than those proposed by Yokoo et al. in [4], namely when an approximate solution may be accepted under a deadline.

Jun Sawamoto and his colleagues [5] proposed a framework to constraint problem solving using a multi agent system approach. The multi agent system is composed of a division-integration agent, a management agent a group of sub-problems agents. The problem in hand is firstly split into a number of sub-problems by sub problems agents after receiving a request from the integrating agent. Sub problems agents then generate sub-plans to resolve the sub problems and ask management agent for resources needed to execute their sub-plans. The management agent notifies resource allocation to the sub problem agents. Then these last generate solutions to sub problems and notify them to the division-integration agent. When this last receive all sub solution to sub problems it integrates them in order to have a global solution to the undertaken problem. The multi agent framework to cooperative problem solving proposed is based on the top down method to handle complex problems but the Management Agent activity in the resource allocation task could be a bottleneck for the whole system, namely when these resources are bounded and shared between agents.

Xuan Nguen and his colleagues [6] proposed a multi agent approach to solve distributed fuzzy problems using asynchronous backtracking algorithm (ABT) [7] which is a variant of the standard backtracking algorithm and where variables are affected by priorities. In this approach agents act in an asynchronously manner and interact by a peer to peer sending messages mechanism, according to prefixed priorities. Although this approach uses the paradigm agent in its two dimensions individual and social, the algorithm used presents a scaling problem [4]. This drawback is notified by the author’s paper himself when he said “However the performance of FABT comes with the cost of complexity in its “nogood storage”. Moreover, it should be said that these three approaches take into account only the case when the relations between domain variables are only quantitative not qualitative ones. According to our knowledge, a multi agent based approach to

ISSN: 1790-5117

qualitative temporal constraint reasoning doesn’t exist in literature up to now. Our approach is intuitive and deals with the whole constraint solving process, including: 1) Qualitative knowledge model construction, which is a temporal constraint qualitative network based on Allen’s interval algebra; 2) knowledge model filtering; 3); seeking of qualitative consistent solutions and 4) visualizations of the results. All these steps are carried out by agents in a cooperative or in a parallel manner (namely the reasoning tasks ones) in the aim to overcome the computational limitations of such interval based relational language as it will be shown in sections 3 and 4.

3 Qualitative temporal constraint reasoning frameworks.

We can distinguish two main frameworks to qualitative temporal reasoning, according to the manner in which temporal knowledge is represented either in interval based framework or in point based one. The former framework is interval based algebra proposed by Allen in [8] to represent and reason about temporal knowledge. Allen’s algebra is based on thirteen (13) basic temporal relations between time intervals corresponding to all the possible ways in which the endpoints of two intervals can be ordered, which are: \{ meets (m), met_by(mi), overlaps(o), overlapped_by(oi), starts(s), started_by(si), during(d) , contains(di), finishes(f), finished_by(fi), before (b), after(a) ,equal(e) \}, the symbol letter between parentheses indicate the temporal relation abbreviation. These basic relations can be combined to derive \(2^{13}\) possible relations. For example, by combining \(A\{a\}B\) and \(A\{o\}B\), we obtain \(A\{a, o\}B\). Composition of this set of relations can be given by a \(13 \times 13\) composition table, for example : \(A\{oi\}B\) composed to \(B\{m\}C\) gives \(A\{o, fi, di\}C\) and as notation we write : \((oi o m) = \{o, fi, di\}\). We give below an illustrative example taken from [2].

Example 1: Let us consider the following situation “Fred was reading the paper while eating his breakfast. He put the paper down and drunk the last of his coffee. After the breakfast, he went for a walk”. This situation could be represented using Allen’s algebra by:

- Paper \{= , d, di, o, oi, s, si, f, fi\} Breakfast; Paper \{d, o, s\} Coffee; Coffee \{d\} Breakfast; Walk \{bi\} Breakfast. Graphically, it can be represented by a qualitative temporal constraint network (Q.T.C.N) depicted in figure 1. To reason on this network, figure 2 gives minimal temporal constraint network and figure 3 gives a consistent scenario.

\[\text{Fig 1: Interval based Q.T.C. N}\]

\[\text{Fig. 2: Minimal Q.T.C.N}\]
The second framework was proposed by [9] and uses point based algebra to represent and reason about temporal knowledge. In this approach entities are events linked by means of temporal relations defined on the basic set: \{<, > , = \}. The combination of the elements of the basic set gives possible temporal relations which are: \{<,>, =, ≤ , ≥, ≠,, ? \}. This framework was proposed as an alternative and a solution to the computational limitations of the interval based framework. Within this framework, it is proven that fundamental reasoning tasks quoted above can be performed in polynomial time [10][11][12]. We give below an illustrative example taken from [2].

**Example 2**: Suppose that we have four events x, y, z, t on which we define the following point based temporal relations (constraints): \(x < y, z < t, x \{>,=\} z, y \{<, =\} t, y > z \). These constraints are represented by the qualitative point based temporal constraints network in figure 4. Figure 5 gives two possible scenarios relating to this temporal network.

4. The proposed Multi agent based approach to qualitative temporal constraint reasoning.

4.1 Motivations of the multi-agent paradigm

Interval based formalism is more expressive than the point based one but unfortunately the former presents computational problems as we have shown in the previous section. All the more, the transformation of a temporal constraints network expressed in the interval based formalism into a point based one is not always possible [2]. This is why we thought of a qualitative constraint reasoning approach which benefits from the expressivity of the interval based formalism while attenuating the reasoning tasks computational time by the use of the multi-agent paradigm which brings into play concepts like communication, cooperation, parallelism and concurrency between agents.

4.2 Multi agent architecture

The multi agent architecture proposed showing the relationships between the different agents is illustrated in Figure 6. This architecture contains a user interface agent, which is the mediator between the user and the system, and four (04) groups of agents where each group has different role. As shown in figure 6, The system can be executed by the user either phase by phase (indicated in figure 6 by 1,2,3,4) or as a whole process (indicated in the figure 6 as 1,5,6,7,4). We give below the description and the roles of these agents.

**a) User Interface Agent (U.I.A)**

Interacts with the user through a graphical user interface (GUI). It subjects to the system the input data to treat and a list of requests resulting from the user which are:

1- Analyzing input data and building the knowledge model.
2- Visualizing the knowledge model
3- Applying arc-consistency to verify the local coherence of the knowledge model
4- Applying path-consistency to verify the coherence of two successive arcs of the knowledge model.
5 - Giving consistent or all consistent scenarios.
6- Visualizing consistent scenarios if found.
7- Executing the whole process (phases 1 to 6).

Also, UIA assumes the task to read the input file and store it in a structured array, visualize scenarios.
(possible qualitative solutions), redirecting the message “no consistent scenario found” to the user. Finally it may also compute and visualize quantitative solutions relating to a scenario, if asked by the user.

b) Data checking Agents (D.C.As)
These agents analyze in a parallel manner the input data and inform the user via the user agent about the anomalies and errors detected in this data, such as:
- Action with an empty interval
- An interval without an action the same action with two different intervals
- An interval whose beginning is higher than the end
- An empty set relation. Appearance of the same action with the same interval more than once between two actions
- Relation and its opposite between the same two actions.

• A new set relations between two intervals of two actions.
• A relation between two intervals I and J different of a yet given invert relation between the same intervals.

c) Qualitative Knowledge Representation Agents (Q.K.R.As)
the input data file contains a succession of structured information composed of the following fields : Action(i), Interval (i) , {C(i,j)},Interval(j), Action(j). This data stored in a structured array by the user agent (See Figure 8). Intervals are imprecise temporal information about the execution of the corresponding actions and the constraints C(i,j) define temporal relations between these intervals.
Given this data, a certain number of knowledge

![Diagram](image.png)

**Fig 4 : Point based Q.T.C.N**

I--------I-----I------------------------ scenario 1.

x=z y t

I--------I-----I------------------------ scenario 2.

z x t=y

**Fig 5: Two Consistent Scenarios**
Fig. 6: The Multi Agent Architecture
representation agents build an adjacency matrix of the qualitative temporal constraint network defined by the input data, in a parallel manner. Each node of the temporal network is represented by an action and its time interval. Nodes are linked to each others by Allen’s qualitative relations as defined in the input data. The number of Q.K.R.As participating to knowledge representation depends on the amount of the input data to be represented and is defined dynamically by the system and destroyed after knowledge model building.

d) Filtering Agents (F.As)

These agents operate on the built qualitative temporal constraint network, represented by its adjacency matrix, in order to detect and to eliminate arc-inconsistencies. An arc-inconsistency is an instantiation of two nodes by a couple of temporal values which is inconsistent with the temporal relation between the two nodes.

The temporal constraints network filtering is carried out by F.As in a cooperative way. We assign to each node a filtering agent so that directly linked agents can communicate and share information about temporal intervals so that they can detect local inconsistencies and eliminate them. The number of filtering agents used depends also on the number of time intervals and are created and destroyed dynamically by the system. To carry out their task, filtering agents use Mackworth’s arc-consistency algorithm [13] depicted in figure 9.

e) Qualitative Problem solving Agents (Q.P.S.As)

These agents cooperate in order to find possible consistent scenarios to the problem represented by the qualitative temporal constraints network. A scenario shows the manner in which intervals are arranged and represents a qualitative solution to the problem. To obtain quantitative solutions relating to a scenario, it is enough to instantiate intervals by numerical values.

Each consistent scenario is obtained using two agents which are path consistency agent (PCA) and consistency agent (GCA). These agents use QPC algorithm and consistency algorithm [14][15], depicted in figures 10 and 11.

4.4 Functionality of the multi agent system

Our multi agent based approach to qualitative constraint reasoning passes by the following phases:

a) Introduction of the input data file by the user to the system via the user interface agent. This data file has the structure : < Action(i), Interval (i) , {C(i,j)},Interval(j), Action(j) > (See Figure 8).

b) Analyzing the input data by certain number of data checking agents in a parallel manner for the cases quoted above. The number of these agents depends on input data file size. These agents are created and destroyed automatically by the system. If any errors or anomalies are detected, these last are declared with the interface agent which in its turn informs the user. The passage of the system to the following phase is not possible so that the input file is free of anomalies.

c) Qualitative representation of input knowledge as a matrix C(N*N) which gives constraints between N*N intervals (Figure 7). This matrix is built in a parallel way by a certain number of qualitative knowledge representation agents which are created and destroyed automatically after the construction of this matrix.

d) Filtering of the matrix representation by filtering agents where each agent is affected to an interval so that each couple of filtering agents share information about linked intervals and thus work in a cooperative way. As quoted above, the filtering operation consists of detecting and eliminating inconsistencies from arcs. It is significant to note that if the operation of filtering produces an empty interval, the agent having in possession of this empty interval informs all the filtering agents of this result and sends an urgent message to the interface agent saying that the problem thus defined does not have any consistent scenario. The filtering agents are created and destroyed dynamically by the system and use the arc consistency algorithm shown in figure 9.
c) Qualitative problem solving by searching for all consistent scenarios. As indicated above, each scenario is obtained thanks to the cooperation of two agents: path consistency agent (PCA) and global consistency agent (GCA) which respectively use algorithms [14][15] of figure 10 and figure 11. Scenarios obtained are stored in a file called scenariosfile. Each time that a scenario is computed and stored, an information message is sent to the interface agent who in its turn informs the user. These agents are created and destroyed dynamically by the system.

f) The visualization of the scenarios stored in scenariosfile can be displayed immediately after their creation or at the end of the entire problem solving stage. Numerical solutions related to a scenario can be calculated and posted within the interface agent after a user request formulation.

5 Implementation
We use JADE platform to implement our approach (Java Agent Development Framework). This choice can be motivated by the fact that this platform is distributed and independent from the domain application and of the hardware architecture used., Moreover, it offers sufficient functionalities and concepts necessary to the implementation of our approach such as the dynamic creation and destruction of agents (Subscribe and Unsubscribe), the cloning of agents and the publication of services (Directory Facilitating) to be offered by the agents. For our case, these services will play the role of various algorithms of which we spoke namely the arc-consistency, path-consistency as well as the search for scenarios and their visualization. Moreover, JADE platform, offers a standard communication language called FIPA-ACL (Foundation of Physical agents – Agent Communication Language) and also various interaction protocols like Request, Inform, Contract.
This procedure examines the arc \((ij)\) and deletes all instantiations of the interval \(I_i\) which are not consistent with the qualitative relation \(C_{ij}\) and \(I_j\) are respectively between nodes \(i, j\). \(C_{ij}\) is the constraint (relation) between nodes \(i\) and \(j\) between nodes \(i\) and \(j\).

******** Procedure \(\text{REVISE}(V_i, V_j, C_{ij})\)  
1. \(\text{del}: = \text{false}\)  
2. if \((V_i, V_j) \in I_i\) is not consistent with \(C_{ij}\) then begin  
3. delete \(V_i\) from \(I_i\)  
4. \(\text{del}: = \text{true}\)  
5. return \(\text{del}\)  

******** Arc consistency algorithm  
AC-algorithm  
** Input: an arc \(X\) with its time intervals \(I_i\) and \(I_j\) and its relation \(C_{ij}\)**  
** Output: a consistent arc \(X\)**  
1. \(Q: = \{ (V_i, V_j) \in X, \ i \neq j\}\)  
2. repeat  
3. change := \text{false}  
4. for each \((V_i, V_j) \in Q\) do  
5. change := ( \(\text{REVISE}(V_i, V_j)\) .OR. change)  
6. endfor  
7. until (not(change))  
8. end

Fig. 9: Arc Consistency algorithm of I.A based constraints networks

******** Path consistency algorithm  
QPC-algorithm  
** input1: constraint matrix \(C(N*N)=\{c_{ij}\}\) verifying arc consistency. **  
** input2: Allen’s composition table (13x13 table) **  
** Output: a matrix \(C(N*N)\) verifying path consistency *****  
1. Repeat  
2. \(S: = C\)  
3. For \(k:=1\) to \(n\) do  
4. for \(i,j:=1\) to \(N\) do begin  
5. \(C_i := C_i \oplus (C_i \otimes C_i)\)  
6. until \(S = T\)  
7. end

Fig. 10: Path consistency algorithm
**Consistency algorithm**

**Input:** Matrix C representing an I.A based set of constraints between N nodes

**Output:** True if the set is satisfiable, false otherwise.

1. \( C := \text{QPC} \ (C) \) \(*\ use \text{QPC}\)
2. if \( C \) contains empty relation then
3. return false
4. else
5. choose an unprocessed relation \( C_{ij} \) and
6. split \( C_{ij} \) into \( R_1, R_2, ..., R_m \)
7. if no relation can be split then
8. return true
9. endif
10. for \( m := 1 \) to \( k \) do
11. \( C_{ij} := R_m \)
12. if consistency\( (C) \) then return true
13. endif
14. endfor
15. return false
16. endif
17. end

**Fig. 11:** consistency checking algorithm of I.A based constraints networks.

net and Propose protocols which permit to us to handle the agents’ interaction. At present time, we began the implementation phase and we’ll wait to have better computational results relating to interval based qualitative temporal reasoning tasks.

6 Conclusion and future work

Allen’s Interval algebra [8] based temporal reasoning is a qualitative temporal reasoning where domain knowledge is modeled by a temporal graph called qualitative temporal constraints network (Q.T.C.N). Each node of a Q.T.C.N is labeled by a domain variable and a time interval and each arc is labeled by a constraint or a subset of constraints belonging to Allen’s temporal constraints (relations) set composed of thirteen (13) basic temporal constraints. Allen’s algebra is expressive but unfortunately, reasoning tasks (arc consistency, path consistency, searching for consistent scenarios, searching for a minimal constraint network) using this language suffers from a computational and scaling problem.

In order to make these reasoning tasks scalable, another qualitative temporal algebra, called point algebra, was proposed by Vilain and Kautz [9]. In the same aim, subclasses of Allen’s algebra were proposed, such as Nebel and Bürckert’s ORD-Horn sub-algebra [16]. Reasoning tasks using point algebra and Allen’s sub-algebras are scalable. Although, their common drawback is being less expressive than Allen’s algebra.

In the objective to mitigate this computational and scaling problem, while benefiting from the entire expressivity of Allen’s temporal language, we proposed a multi agent based approach which conceives a problem solving as the interaction of a certain number of autonomous entities called agents. This interaction is implemented by communication, cooperation, parallelism and competition between agents. As we have mentioned
it in the section 2, we haven’t found in the literature any other multi-agent approach to qualitative constraint reasoning and our approach may be considered as the first essay.

Nevertheless, the proposed approach remains opened to future work enumerated below:

1) Initially, to complete the implementation of our approach and to evaluate the results obtained for different data sizes (small, large and very large data).

2) Our approach can return only complete solutions (if they exist) which verify all the given constraints. If a complete solution does not exist, it would be very interesting that a future extension of our work can propose an approximate optimal solution verifying the maximum number of defined constraints.

3) Taking into account the case where the input data is introduced from different information sources. In this case, two possibilities are to be considered:

   a- Locally building the qualitative constraint networks then to integrate them and finally reasoning on them in a parallel manner.

   b- Locally building the qualitative temporal constraint networks then to locally reason on them in order to seek the local solutions and finally to integrate them thereafter.

References


