A Visualized Framework for Representing Uncertain and Incomplete Temporal Knowledge

Yue Wang

A dissertation submitted in partial fulfilment of the requirements of the University of Greenwich for the degree of Doctor of Philosophy

August 2014

The University of Greenwich,

School of computing and Mathematical Science,

Park Row, Greenwich, SE10, 9LS

DECLARATION

I certify that this work has not been accepted in substance for any degree, and is not currently submitted for any degree other than that of Doctor of Philosophy being studied at the University of Greenwich. I also declare that this work is the result of my own investigations except where otherwise identified by references and that I have not plagiarized the work of others.

X---------------------------------

Yue Wang

X---------------------------------

Dr. Jixin Ma

(Supervisor)

X---------------------------------

Dr. Alan Soper

(Supervisor)

ACKNOWLEDGEMENTS

Firstly, I am very grateful to Dr Jixin Ma and Prof. Huaikou Miao for creating the opportunity for me to come to University of Greenwich to fulfil my dreams of pursuing a PhD project here. Especially, I would like to thank my supervisor Dr Jixin Ma and Dr Alan Soper for their continual supporting and encouragement during the three years it took to produce this work.

I am especially grateful to professor Brian Knight, Dr Aihua Zheng, Dr Guoxin Zhao and Dr Xiaoyi Zhou, who shared their precious insights without reservation throughout my research work.

I owe my loving thanks to my parents and my wife for their constant support and understanding in all my professional endeavours.

The financial support of the University of Greenwich is gratefully acknowledged.

ABSTRACT

This thesis presents a visualized framework, called Visual Time, for representing and reasoning about incomplete and uncertain temporal information. It is both expressive and versatile, allowing logical conjunctions and disjunctions of both absolute and relative temporal relations, such as “Before”, “Meets”, “Overlaps”, “Starts”, “During”, and “Finishes”, etc. In terms of a visualized framework, Visual Time provides a user-friendly environment for describing scenarios with rich temporal structure in natural language, which can be formatted as structured temporal phrases and modelled in terms of Time Relationship Diagrams (TRD). A TRD can be automatically and visually transformed into a corresponding Time Graph, supported by automatic consistency checker that derives a verdict to confirm if a given scenario is temporally consistent or inconsistent. The thesis provides the following contributions:

1. **Extended graphical representation for uncertain and incomplete temporal knowledge:** An extended graphical representation for uncertain and incomplete temporal knowledge based on [KM1992] is proposed, supporting both logical connectives ‘$\wedge$’ and ‘$\vee$’. In Chapter 3, it is shown all the other logical connectives can be derively defined.

2. **Time relation diagram (TRD):** A time relation diagram (TRD) is designed for representing temporal relations between time elements which could be both point and interval. Each time element is denoted as a box consisting of three components: Name, Duration and Property. Temporal relations are denoted in terms of directed arcs. TRD allows expressions of both absolute and relative temporal relations, supporting both logical conjunctions and disjunctions.

3. **A semi-automatic temporal information extractor:** SUTime is a very useful tool for extracting verbs and temporal information [CM2012]. However, the extracted verbs and temporal information may play different roles when modelled by TRD. For example, in "*He starts to start the car*", "*start*" is an event while "*starts*" means the action "*start*" happens. An improved algorithm called Temporal Extractor algorithm (TE) is introduced in Section 4.2. Based on Stanford SUTime, TE can semi-automatically extract time elements and temporal relations from any arbitrary text to create a TRD.

4. **Four algorithms:** The first algorithm, Temporal Relation Algorithm (TRM), is designed to extract temporal relations from TRD. The second algorithm, Meets Table Algorithm (MTM) is introduced to convert all the extracted temporal relations into a Meets table. The third algorithm, Time Graph Algorithm (TGM) is described to draw the corresponding time graph of a given TRD. The fourth algorithm, Consistency Checking Algorithm (CCM), is designed to check the consistency of TRD. If the TRD is inconsistent, an audio verdict will alert and the corresponding time element(s) and natural texts will be marked in red colour.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

CC	coordinating conjunction
CD	cardinal number
DT	determiner
EX	existential there
FW	foreign word
IN	preposition/subordinating conjunction
JJ	adjective
JJR	adjective, comparative
JJS	adjective, superlative
LS	list marker
MD	modal
NN	noun, singular or mass
NNS	noun plural
NNP	proper noun, singular
NNPS	proper noun, plural

PDT	predeterminer
POS	possessive ending
PRP	personal pronoun
PRP$	possessive pronoun
RB	adverb
RBR	adverb, comparative
RBS	adverb, superlative
RP	particle
TO	to
UH	interjection
VB	verb, base form
VBD	verb, past tense
VBG	verb, gerund/present participle
VBN	verb, past participle
VBP	verb, sing. present, non-3d
VBZ	verb, 3rd person sing. present

WDT	wh-determiner
WP	wh-pronoun
WP$	possessive wh-pronoun
WRB	wh-abverb
TRD	Time Relation Diagram
NLP	Natural Language Processing
BMP	Business Process Model

CHAPTER 1 INTRODUCTION

Every concept of time arises in the context of some (no doubt useful) human purpose and bears, inevitably and essentially the stamp of that human intent.

N. Lawrence [1978, p. 24]

Time is very important in our daily life. If a boy in a bar wants to know the marital status of the attractive girl beside him, he may start the conversation by saying: "Don't drink too much, time is getting late, your husband will worry about you.” Actually, this guy is waiting for the answer - "Thanks, but I am single and lonely". However, time not only can be used to strike up a conversation, but can also be used to model all the events that occur over the world.

The study of time spans a variety of different disciplines, such as physics, philosophy, literature, computer science, etc. Starting from the middle of the last century, temporal logic has been tensely studied in the domain of Artificial Intelligence for representing, and reasoning about, propositions qualified in terms of time. However, researchers found that temporal logic could be applied in a wide range of science or humanities projects. The visualization of temporal information is a recent research field that has attracted researchers from many fields such as human-computer interaction(HCI), databases, medical informatics, multimedia, and the new specific field of Information Visualization(IV) [CMS1999] [Chi2001] [CPP1999] [CK1991] [KM2001] [PMR1996] [SC2000].

The works with regard to temporal modelling contribute on the graphical representation of uncertain and incomplete temporal information, temporal reasoning and temporal consistency checking.

Section 1.1 The Aim and the Problems

Uncertain or Incomplete Temporal Information (UITI) has been extensively used in many fields such as economics, video game, and medical service and so on [Cha2014] [FT1993] [EDE2012]. Various approaches to dealing with uncertain or incomplete temporal knowledge have been proposed, but most of them actually devoted themselves to specific applications [IL1994] [Tod1993] [SIB2010] [MLM2010].

The main aim of this research project is to develop a visualized framework for representing and reasoning about uncertain and incomplete temporal knowledge.

1. Graphical representation of uncertain and incomplete temporal knowledge

To graphically representing uncertain and incomplete temporal knowledge, there are two problems:

1) How to graphically represent uncertain and incomplete temporal knowledge?

This leads to the following questions:

i. How to express all the possible uncertain and incomplete temporal knowledge?

ii. How can logic connectives ‘∧’ and ‘∨’ be used to express all the possible uncertain and incomplete temporal knowledge?

2) After graphically representing uncertain and incomplete temporal knowledge, how to construct a reliable method of inference, based on this representation?

Two sub-questions are:

i. How to check the consistency of given temporal knowledge?

ii. How to express the consistency result?

2. Visually modelling temporal knowledge

Temporal information can be absolute or relative such as “1:00 PM” and “after the world war”. The problems of modelling temporal knowledge are shown as follows:

1) What are the main attributes of a time element?

2) How to express the uncertainty and incompleteness in the time relation diagram?

3. Extracting temporal information from natural text

As a sub-field of AI research, temporal logic can be used to model the events happening in daily life. Social media has become an important information source for collecting temporal information. People, especially the young, are more likely to post their pictures and ideas on social media.

Figure 1.1 shows three important pieces of information as follows:

- This social media user likes Greenwich.
- The posted time of this tweet is 10:04 AM on 13 January 2014.
- The location of the user sending this tweet is Greenwich, London.

Figure 1.1 Social Media Example

On 16 April 2013, the Boston Marathon was hit deadly by twin explosions. Public security officers and FBI collected plenty of data from the internet and social media websites according to its time and location, and this data played a key role in solving that case. However, the data collected from social media was large, unfiltered and unmediated. Graphic images of the injured flooded Twitter, as did inaccurate or downright false reports. Today, the information we can obtain is increasing quickly (2.5 Exabyte (2.5×10^{18}) of data was created every day in 2012). However, researchers have to face two new challenges:

1) How to extract temporal information from natural text?

2) How to model the extracted temporal information?

The research on question 1) can help the police officers to collect the criminal information and can help the public health sector analysing the topical illnesses people care about. The research on question 2) will help with Case-Based Reasoning (CBR), temporal database, multi-agent systems.

Section 1.2 The Motivation and Objectives

The motivation of this research project is to develop a Visualized Framework, where, firstly, this framework can be used to model temporal knowledge directly from natural language texts. Users can manually input the natural language texts in this framework or import temporal information from a given text document. Users can also use an online social media filter to download temporal information from internet.

Secondly, up to now, many temporal models have been put forward such as temporal models for multimedia synchronization [li1997] and temporal models of pitch processing [Yos1999]. Most of these works have not been visualized and therefore not straightforward for non-expert users. To solve these problems, a Temporal Relationship Diagram (TRD) is designed which gives a systematic way of describing and defining temporal knowledge, which is from the idea of Peter Chen's Entity-relationship model

[Che1976]. Temporal knowledge is modelled as components that are linked with each other by temporal relations that express the dependencies and requirements between them. TRDs are created to represent these time elements, durations (time length), and temporal relationships graphically.

Thirdly, another motivation of this thesis is to design an automatic consistency checker, which can show the consistency results directly on the original natural texts, instead of just giving the user the consistency result or some mathematical logic expressions. Three kinds of mathematical results (meets table, temporal relations and time graph) will also be needed for the future works. Time relation diagram (TRD) is put forward in this thesis, TRD is a model designed for representing the time relations among time elements, which could be both point and interval. TRD can be automatically and visually transformed into its corresponding Time Graph by this framework, which will deliver a visual and audio verdict as to whether a given scenario is temporally consistent or not. If it is inconsistent, the consistency result will visually pick out the original natural texts that make this knowledge inconsistent.

Section 1.3 The Contributions

This thesis contributes to the area of temporal representation and visualization in the domain of Artificial Intelligence. A framework for representing uncertain and incomplete temporal knowledge is proposed in terms of a visualized case tool called Visual Time. Four main contributions are summarized as follows:

1. **Extended graphical representation for uncertain and incomplete temporal knowledge:** An extended graphical representation for uncertain and incomplete temporal knowledge based on [KM1992] is proposed, supporting both logical connectives ‘$\wedge$’ and ‘$\vee$’. In Chapter 3, it is shown all the other logical connectives

can be derively defined.

2. **Time relation diagram (TRD):** A time relation diagram (TRD) is designed for representing temporal relations between time elements, which could be both point and interval. Each time element is denoted as a box consisting of three components: Name, Duration and Property. Temporal relations are denoted in terms of directed arcs. TRD allows expressions of both absolute and relative temporal relations, supporting both logical conjunctions and disjunctions.

3. **A semi-automatic temporal information extractor:** SUTime is a very useful tool for extracting verbs and temporal information [CM2012]. However, the extracted verbs and temporal information may play different roles when modelled by TRD. For example, in "*He starts to start the car*", "*start*" is an event while "*starts*" means the action "*start*" happens. An improved algorithm called Temporal Extractor algorithm (TE) is introduced in Section 4.2. Based on Stanford SUTime, TE can semi-automatically extract time elements and temporal relations from any arbitrary text to create a TRD.

4. **Four algorithms:** The first algorithm, Temporal Relation Algorithm (TRM), is designed to extract temporal relations from TRD. The second algorithm, Meets Table Algorithm (MTM) is introduced to convert all the extracted temporal relations into a Meets table. The third algorithm, Time Graph Algorithm (TGM) is described to draw the corresponding time graph of a given TRD. The fourth algorithm, Consistency Checking Algorithm (CCM), is designed to check the consistency of TRD. If the TRD is inconsistent, an audio verdict will alert and the corresponding time element(s) and natural texts will be marked in red colour.

Section 1.4 The Expected Impact and Applications

1. **Extended graphical representation for uncertain and incomplete temporal knowledge:** To graphically represent temporal knowledge, this thesis puts forward an extended time graph theory that can help to represent any kind of visualization on temporal information based on point and interval based system.

2. **Time relation diagram (TRD):** TRD can be used to model temporal scenarios, in which, all the time elements, their relations and durations (time length) can be modelled graphically as a diagram. It can help to build temporal database and conduct temporal reasoning.

3. **A semi-automatic temporal information extractor:** Beginning in 1987, information extraction (IE) was spurred by a series of Message Understanding Conferences (MUC 1 - MUC 7). This extractor is designed to extract temporal information directly from natural language, which will be modelled by TRD. It can help in building temporally aware systems and investigating problems requiring temporal information, such as event extraction, temporal ordering of events, and question answering.

4. **Four algorithms:** Temporal Relation Algorithm (TRM), Meets Table Algorithm (MTM) and Time Graph Algorithm (TGM) will help to build the corresponding Time relations, Meets table and time graph of TRD. Consistency Checking Algorithm (CCM) will help to check the consistency of TRD. If TRD is inconsistency, the result shown in natural language will help to check which natural sentence(s) causes TRD inconsistency. The result shown in TRD can be edited by the expert to test how to make it consistent.

Section 1.5 Thesis Structure

Chapter 2 will critically introduce all the background theory of the framework. In the temporal basis section, point based system, interval based system and point and interval based system will be introduced and critically analysed. In Section 2.2, two natural language tools: Stanford Parser and SUTime are introduced and critically discussed. At the end of Chapter 2, a summary conclusion is given to introduce relationship between this thesis and the related literature.

Chapter 3 will introduce an extended graphical representation of uncertain and incomplete temporal knowledge, and the consistency of uncertain and incomplete temporal knowledge is discussed.

Chapter 4 is the main body of this thesis, where a visualized case tool for uncertain or incomplete temporal knowledge called Visual Time is specifically introduced. The Bilingual Online Search Engine, Semi-automatic natural language extractor, Time Relation Diagram (TRD), Meets Table algorithm, Time Graph algorithm and Consistency Checking algorithm are comprehensively introduced.

Chapter 5 will use a scenario with a background of a virtual terrorist attack happening in University of Greenwich to express how the framework works. Some examples collected from Twitter or Sina Weibo are used for critical testing.

In chapter 6, a summary of the current research in this thesis is outlined and the concluding recommendations on the outcome of the research are proposed. The contributions are restated.

In chapter 7, suggested future works are presented.

There are also two appendixes for this thesis. Appendix A is the related C# programming source codes. Appendix B shows the speech tag of the conventions of the first-order predicate calculus with equality throughout.

CHAPTER 2 CRITICAL LITERATURE RIVIEW

Arthur Prior founded tense logic [Art1957] [Art1967][Art1968], now also known as temporal logic, and made important contributions to intentional logic. Arthur Prior illustrated that this discipline was closely related to model logic, especially to express and model human temporal concepts in a natural way. Even though Prior preferred to use the term "tense logic", Amir Pnueli [Ami1977] and other logicians developed Prior's theory and named it "Temporal logic". Recent researchers found that as most of temporal information collected from daily life kept changing, it was impossible to use the old temporal logic theory to model real time events; this means that a new temporal system is required to deal with uncertain or incomplete temporal knowledge.

At first, researchers used temporal logic to reason about a time line in what were called Linear Time Logics. After that, researchers found that sometimes the environment may act unpredictably, the information they got may be uncertain and incomplete, and then Branching Logics was introduced to reason about multiple time lines. Quite a lot has been written on the temporal logic, which was called tense logic in the early ages. In recent years, most of published works are concentrating on specific applications such as databases, planning methods, sensor networks and so on [IL1994] [Tod1993] [SIB2010] [MLM2010].

Some commonly logical connectives and logical operators are shown as follows:

1. Negation (not): ¬

2. Conjunction (and): $\wedge$

3. Disjunction (or): $\vee$

4. Material implication (if...then): $\Rightarrow$

5. Biconditional (if and only if): $\Leftrightarrow$

6. Exclusive OR: ∇

7. For all : $\forall$

8. There exists: $\exists$

9. Is an element of: $\in$;

10. Is a subset of : $\subset$.

Section 2.1 Temporal Logic

If everything when it occupies an equal space is at rest, and if that which is in locomotion is always occupying such a space at any moment, the flying arrow is therefore motionless.

– As recounted by Aristotle, Physics VI:9, 239b5

There is a long-standing debate about the question of what sort of objects should be taken as time elements? However, "point or interval" is a very old question. The above scenario is known as Zeno's paradoxes, which reflects that human have a natural desire to pursue knowledge of the nature of time. In many temporal systems, there are some fundamental differences in the time ontology. Up to now, there are three known ideas:

1. Points (instants of time with no duration) [Bru1972] [Sho1987] [Lad1987] [HS1991] [Lad1992];

2. Intervals (periods of time with positive duration) [All1983] [All1984] [AH1989];

3. Both points and intervals [MK1994] [MK1996] [MK2003] [MH2006].

The key difference between the above three theories is whether intervals should be taken as primitive or not. [MH2006] argued that they are actually reducible to logically equivalent expressions, under some requisite interpretations, and therefore they can also be viewed as ally.

Section 2.1.1 Point-based Systems

Researchers tried to argue that every event must have a start time and an end time, and sometimes the start time and the end time are the same one, for example, "*The plane will depart at 2:00 PM, and will reach its destination at 4:00 PM*", "*The database will update at 2:00PM*." A typical point-based time structure is an ordering (P,≤), where P is a set of points, and ≤ is a relation that (partially or totally) orders P. In point-based systems, intervals may be defined as derived temporal objects, either as sets of points in [Dre1982] [Ben1983], or as orderings of points in [Bru1972] [Sho1987] [Lad1987] [HS1991] [Lad1992].

A point-based system is so useful in some aspects that there are still some application papers which have been published in recent years [Bru1972] [FB2000][GS2010]. This thesis will use Bruce's model [Bru1972] to illustrate this theory.

The motivation of Bruce's model [Bru1972] is to solve a natural language processing problem - Question Answering Program (named Chronos). However, in [Bru1972], Bruce argued that even though quite a lot of papers have been written on the phenomena of time and tense, much of these works at that era has been done with little effort towards integrating it with other work. There are three goals in Bruce's paper. First, like any formal system, it should be precise, well defined, and internally consistent. It should be general enough for a variety of applications, but specific enough to reflect the full structure underlying temporal expressions. It should be a systematic development of a complex structure from a few primitive notions. Second, the model should serve as a common framework for discussion of tenses from the linguist's viewpoint or from that of the tense logician. It should demonstrate the relationship of the various tenses to one another and to other time relations. Third, it should be an accurate representation of the intuitive meaning of temporal expressions.

Based upon first-order (also known as first-order predicate calculus) Bruce tried to design a framework for analysis of tenses, time relations, and other references to time in language. Bruce gives seven binary relations (See Figure 2.1) between time-segments (point-based temporal intervals) [Bru1972], which can be derived from the ordering relations over their greatest lower bounds and the least upper bounds: Before, During, Same-time Overlaps, After, Contains and Overlapped. A tense is defined as a special n-ary relation on time-segments with the following form:

$$R_1R_2 \ldots R_{n-1}(S_1, S_2 \ldots S_n) \equiv R_1 (S_1, S_2) \wedge R_2 (S_2, S_3) \wedge \ldots \wedge R_{n-1}(S_{n-1}, Sn)$$

where each S_i is a time-segment and R_i is a binary relation between S_i and S_{i+1}. S_1 is called *time of speech*, and S_2,..., S_n are called time references, and S_n is called the time of *event.*

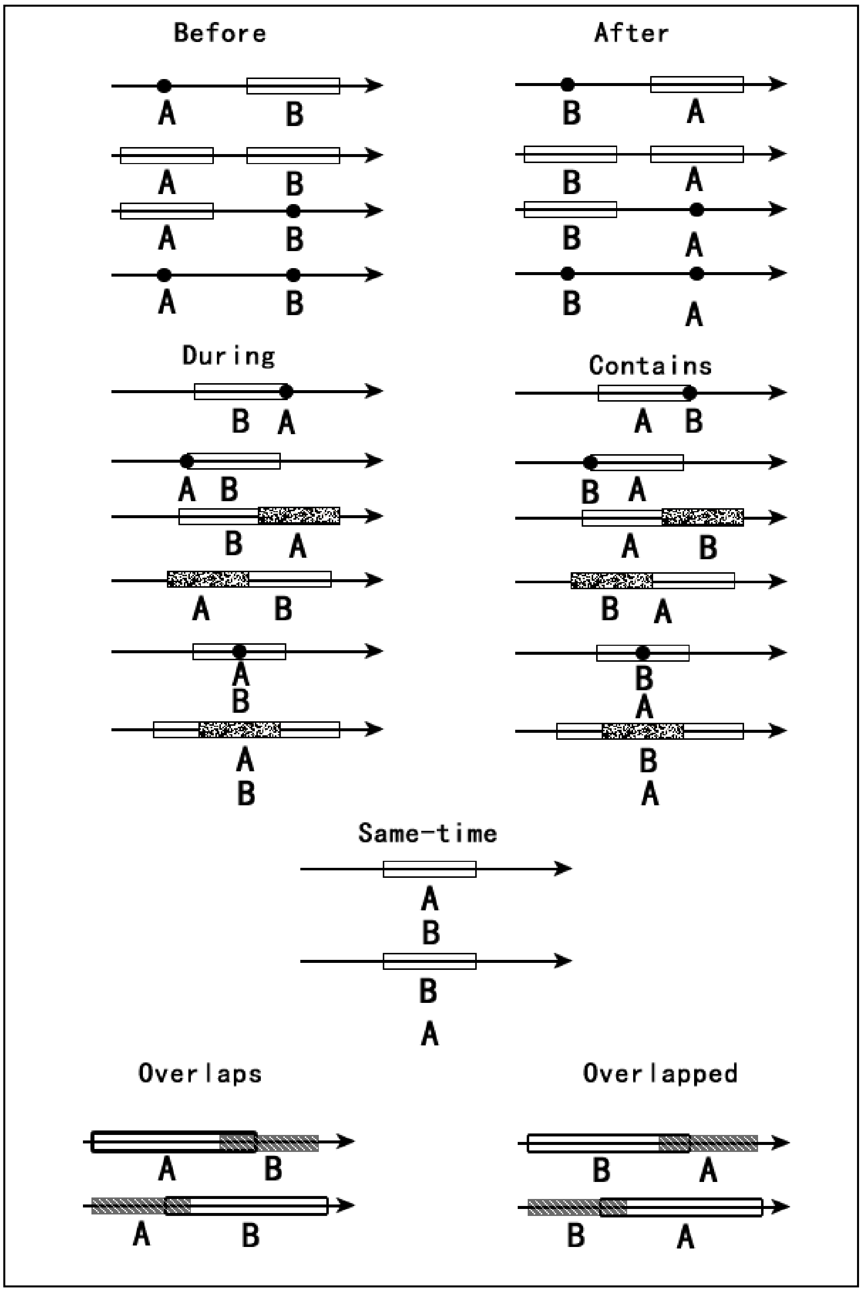


Figure 2.1 Bruce's Seven Binary Relations

For example, Bruce used one natural language sentence as follow:

He will have been going to be going to go has the tense

$$Before\_After\_Before\_Before(S_1,S_2,S_3,S_4,S_5) \equiv$$
$$Before(S_1,S_2) \wedge After(S_2,S_3) \wedge Before(S_3,S_4) \wedge Before(S_4,S_5)$$

Where S_i is the time of speech, S_2, S_3, and $S4$ are reference times, and S_5 is the time of event.

It has been argued by some researchers that defining intervals as objects derived from point may lead to the so-called Dividing Instant Problem [AH1989] [Bru1972] [Lad1987], which is in fact an ancient historical puzzle encountered when attempting to represent what happens at the boundary point that divides two successive intervals.

For example, consider the two states, “The light was off” and “The light was on” which hold true throughout two successive point-based intervals, say $<p_1, p>$ and $<p, p_2>$, respectively; then the question becomes as “Was the light off or on at point p?” This, in terms of the *open* or *closed* nature of the involved point-based intervals, turns out to be the question of which of the two successive intervals, i.e., $<p_1, p>$ and $<p, p_2>$, is closed/open at the dividing point p? Practically, there are four possible cases:

(a) The light was off rather than on at p;

(b) The light was on rather than off at p;

(c) The light was both off and on at p;

(d) The light was neither off nor was it on at *p*.

While both (c) and (d) are illogical, since the former claim violates the *Law of Contradiction* and the latter violates the *Law of Excluded Third* [Ben1983], the choice between (a) and (b) must be arbitrary and artificial. In fact, since we have no better reason, from the point of view of philosophy, for saying that the light was off than for saying that it was on at the dividing-instant, such an arbitrary approach has been criticized as indefensible and hence unsatisfactory [Ben1983] [All1984] [Gal1990] [Vil1994].

Conclusion: Comparing to the related works [Bul1962] [Had1967][Jam1970], Bruce's point-based system is the first work to conclude seven temporal relations: Before, During, Same-time, Overlaps, After, Contains and Overlapped which play a key role to affect the related research filed. Most the related works such as [All1984] [Ma1994] [Ben1983] [Gal1990] tried to extend these temporal relations. However, to overcome the dividing instant problem, a further work called interval-based system was proposed by [All1984] which will be introduced in Section 2.1.2.

Section 2.1.2 Interval-based Systems

The point-based structure of time has been challenged by many researchers who believe that time intervals are more suited for the expression of common sense temporal knowledge, especially in the domain of linguistics and AI [All1983] [All1984] [AH1989]. Then the interval-based system was proposed. A typical interval-based time structure is an ordering (***I***, ***R***), where ***I*** is a set of intervals, and ***R*** is a set of binary relations over ***I*** [All1984]. However, this thesis will use Allen and Hayes' approach to explain interval-based systems

[AH1989].

The nature of Allen's intervals theory is that only intervals were taken as the temporal primitives, completely excluding time points from the temporal ontology [All1983] [All1984]. The interval-based theory starts with one primitive object, the time period, and one primitive relation: ***Meets.***

A time period intuitively is the time associated with some event occurring or some property holding in the world. Intuitively, two periods m and n meet if and only if m precedes n, yet there is no time between m and n, and m and n do not overlap with each other [AH1989]. Also there is no beginning or ending to time and there are no semi-infinite or infinite periods.

Allen gave five definitions to restrict periods (temporal interval).

1) First, there is no beginning or ending to time and there are no semi-infinite or infinite periods. This means every period has a period that meets it and another that it meets:

$$\forall i.\ \exists j,k. Meets(j,i) \land Meets(i,k)$$

2) Second, periods can compose to produce a larger period. For any two periods that "Meets", there is another period that is the "concatenation" of them:

$$\forall i, j, k, l.Meets(i, j) \wedge Meets(j, k) \wedge Meets(k,l) \supset \exists m.Meets(i, m) \wedge Meets(m, l)$$

Where Allen used "j + k" to denote the interval that is the concatenation of intervals ***j*** and ***k***, this functional notation is justified because we can prove that the result of ***j + k*** is unique [AH1989].

3) Periods uniquely define an equivalence class of periods that meet them:

$$\forall i, j, k, l.Meets(i, j) \wedge Meets(i, k) \wedge Meets(l,j) \supset Meets(l, k)$$

Where if *i* meets *j* and *i* meets *k*, then any period *l* that meets *j* must also meet *k.*

4) These equivalence classes also uniquely define periods:

$$\forall i, j, k, l.Meets(k, j) \wedge Meets(k, i) \wedge Meets(i,l) \wedge Meets(j, l) \supset i = j$$

Where *i* and *j* both meet the same period *l*, and another period *k* meets both of *i* and *j*, then *i* and *j* are equal.

5) For any two pairs of periods, such that ***i*** meets ***j*** and ***k*** meets ***l***, then either they both

meet at the same "place", or the place where ***i*** meets ***j*** precedes the place where ***k*** meets ***l***, or vice versa.

$$\forall i, j, k, l.(Meets(i, j) \wedge Meets(k, l)) \supset$$

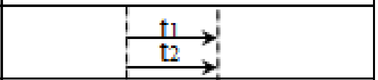
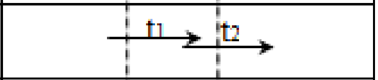
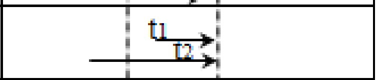
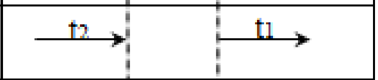
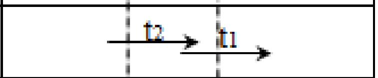
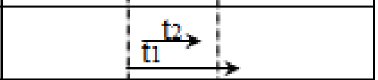
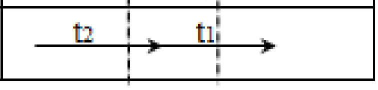
$$Meets(i, l) \oplus (\exists m.\ Meets(k,m) \wedge Meets(m,j)) \oplus (\exists m.\ Meets(i,m) \wedge Meets(m,l))$$

Afterward, Allen argued that he could use the relation "Meets" to describe all the seven binary relations. For example:

$$Before\ (t_1, t_2) \Leftrightarrow \exists t \in T.\ (\ Meets(t_1, t) \wedge Meets(t, t_2))$$

Based on Bruce's seven binary relations, Allen put forward thirteen time relations (Equal, Before, Meets, Overlaps, Starts, During, Finishes, Finished-by, Contains, Started-by, Overlapped-by, Met-by, After) which are shown in Table 2.1.

Table 2.1 Allen's thirteen temporal relations

Relation	Characterized (axiom)	Graph
Equal(t_1, t_2)	$\exists t', t'' \in \mathbf{T}(Meets(t', t_1) \wedge Meets(t', t_2) \wedge Meets(t_1, t'') \wedge Meets(t_2, t''))$	
Before(t_1, t_2)	$\exists t \in \mathbf{T}(Meets(t_1, t) \wedge Meets(t, t_2))$	
Overlaps(t_1, t_2)	$\exists t, t_3, t_4 \in \mathbf{T}(t_1 = t_3 \oplus t \wedge t_2 = t \oplus t_4)$	
Starts(t_1, t_2)	$\exists t \in \mathbf{T}(t_2 = t_1 \oplus t)$	
During(t_1, t_2)	$\exists t_3, t_4 \in \mathbf{T}(t_2 = t_3 \oplus t_1 \oplus t_4)$	
Finishes(t_1, t_2)	$\exists t \in \mathbf{T}(t_2 = t \oplus t_1)$	
After(t_1, t_2)	Before(t_2, t_1)	
Overlapped-by(t_1,t_2)	Overlaps(t_2, t_1)	
Started-by(t_1, t_2)	Starts(t_2, t_1)	
Contains(t_1, t_2)	During(t_2, t_1)	
Finished-by(t_1,t_2)	Finishes(t_2,t_1)	
Met-by(t_1,t_2)	Meets(t_2,t_1)	

However, in Allen's theory, there is only interval in an interval-based system. To avoid the so-called Dividing Instant Problem, Allen and Hayes revised this theory in [AH1989]. In the new theory, to characterize the times that some *"instant-like"* events occupy, they introduce a new kind of interval - ***moment*** whose time length is so short that we can see it as a point.

One limitation of Allen and Hayes' time theory is that it takes only intervals, rather than points, as primitive time elements, although points are later introduced as the "meeting places" of intervals at a subsidiary status within the theory. Their contention is that nothing can be true at a point, for a point is not an entity at which things happen or are true. However, as Galton shows in his critical examination of Allen's interval logic [Gal1990], the theory of time based on intervals is not adequate, as it stands, for reasoning correctly about continuous change. We may illuminate the problem involved with reference to time points by means of the following example of a ball thrown vertically into the air: The motion may be described qualitatively by the use of two intervals, interval i where the ball is going up, and interval j where the ball is coming down. According to classical physics, there is a point p at which the ball is stationary.

As Allen suggested, in the interval calculus, there are two alternatives: we may assume that there is a very small interval where the ball is stationary, or we may assume that interval i "meets" interval j. The first alternative does not seem tenable, being inconsistent with the laws of physics, no matter how small the interval. The second alternative also gives problems, since the interval calculus allows us to combine two intervals which meet, that is, i + j = k [All1984][AH1989]: in Allen's logic, the formula HOLDS(pro, I) is used to say that the property pro holds during the interval I. More precisely, what it says is that pro holds throughout that interval [Gal1990]. However, although the property "ball_in_motion" holds throughout both of intervals i and j, that is:

HOLDS(ball_in_motion,i),

HOLDS(ball_in_motion,j)

One cannot assert that

$$HOLDS(ball\_in\_motion, i+j),$$

since the property "*ball_in_motion*" does not hold throughout the whole combined interval k, within which there is a point p at which the ball is stationary.

To characterise the times that some "instant-like" events occupy, Allen and Hayes introduce the idea of *very short intervals*, called *moments*. A moment is simply a non-decomposable time interval. The important distinction between moments and points is: although being non-decomposable, moments are defined by having extent and by means of having distinct beginning and end points (just as for other intervals [AH1989]), while points are defined by having no extent.

Relating to the meets relation, another obvious difference between points and moments is that moments can meet other intervals, and hence stand between them, while points are not treated as primitive objects and cannot meet anything. However, as Allen and Hayes [AH1989] themselves point out, a theory incorporating granularity involves introducing a "*tolerance relation*" that defines when two times are indistinguishable. For example, two intervals, i and j, might be indistinguishable if their beginning points are at most a moment apart, and likewise for their end points. To ensure that the tolerance relation is an equivalence relation, Allen and Hayes propose axiom <AA>, which insists that moments never meet:

$$\langle AA \rangle \ \forall m,n \in I(moment(m) \wedge moment(n) \Rightarrow \neg meets(m,n))$$

where *moment*(m) is defined by:

$$\forall m \in I(moment(m) \Leftrightarrow \neg \exists i,j \in I(m = i + j))$$

Allen and Hayes declare that their formulation permits either discrete or continuous time models, as well as more exotic models that may alternate between continuous and discrete stretches of time. Unfortunately, axiom <AA> leads to another limitation to the primitive time elements: for any interval, either it is non-decomposable, that is, a moment, or it must be infinitely decomposable. For if it is only finitely decomposable, then it must be the sum of a finite number of moments which would meet one another, contrary to <AA>. This precludes discrete models from the theory containing axiom <AA>. In addition, dense models of the theory, i.e. where all intervals are infinitely decomposable, permit no moments at all, so that <AA> is only vacuously true. Hence models of the theory including <AA> which contain moments can be neither dense nor discrete.

However, although <AA> appears to bring little benefit in the form that is presented here, dealing with moments, it is shown in the next section to play a critical role in a general theory if it is applied to "*time points*". In this case the axiom does not limit the interval structure at all.

Conclusion: Allen claims in his papers [All1981] [All1983] [All1984] that, an interval-based approach avoids the annoying question of whether or not a given point is part of, or a member of a given interval. Allen’s contention is that nothing can be true at a point, for a point is not an entity at which things happen or are true. Therefore, this point of view can successfully overcome/bypass puzzles like the Dividing Instant Problem. However, because there are some problems in this system which cannot be expressed and explained [Gal1990] [MK1994], a further work called point and interval-based system was proposed in [MK1994] which will be introduced in Section 2.1.3.

Section 2.1.3 Point And Interval-based Systems

In order to overcome the limitations of an interval-based approach while retaining its convenience of expression, Point and Interval Based System (PIBS) has been introduced in [MK1994], which addresses both points and intervals as temporal primitives on an equal footing: points do not have to be defined as limits of intervals and intervals do not have to be constructed out of points. In other words, point is point and interval is interval. PIBS is an extension of Allen and Hayes' axiomatisation by the addition of axioms relating to the inclusion of time points as primitive elements.

The time theory, T, takes a nonempty sort, *T*, of primitive time elements, with a primitive order relation 'Meets' over time elements, and a function 'Dur' ('Dur' means time length of time element) from time elements to non-negative real numbers. The basic set of axioms concerning the triad (*T*, Meets, Dur) is given as below:

A1. $\forall t_1,t_2,t_3,t_4 \in \boldsymbol{T}(meets(t_1,t_2) \wedge meets(t_1,t_3) \wedge meets(t_4,t_2) \Rightarrow meets(t_4,t_3))$

That is, if a time element meets two other time elements, then any time element that meets one of these two must also meet the other. This axiom is actually based on the intuition that the "place" where two time elements meet is unique and closely associated with the time elements [AH1989].

A2. $\forall t \in \boldsymbol{T} \exists t',t'' \in \boldsymbol{T}(meets(t',t) \wedge meets(t,t''))$

That is, each time element has at least one immediate predecessor, as well as at least one immediate successor.

$$\text{A3.}\ \forall t1,t2,t3,t4 \in \boldsymbol{T}\ (Meets(t1, t2) \wedge Meets(t3, t4) \Rightarrow$$

$$Meets(t1, t4)\ \nabla\ \exists t'(Meets(t1, t') \wedge Meets(t', t4))\ \nabla\ \exists t''(Meets(t3, t'')$$

$$\wedge\ Meets(t'', t2)))$$

where ∇ stands for "exclusive OR". That is, any two meeting places are either identical or there is at least a time element standing between the two meeting places if they are not identical.

$$\text{A4.}\ \forall t1,t2,t3,t4 \in \boldsymbol{T}\ (Meets(t3, t1) \wedge Meets(t1, t4) \wedge Meets(t3, t2) \wedge Meets(t2, t4))$$

$$\Rightarrow t1 = t2)$$

That is, the time element between any two meeting places is unique.

N.B. In this thesis, for any two adjacent time elements, that is time elements t_1 and t_2 such that Meets(t_1, t_2), we shall simply use $t_1 \oplus t_2$ to denote their ordered union. The existence of such an ordered union of any two adjacent time elements is guaranteed by axioms A2 and A3, while its uniqueness is guaranteed by axiom A4.

A5. $\forall t_1,t_2 \in \boldsymbol{T}$ (Meets(t_1, t_2) $\Rightarrow$ Dur(t_1) > 0 $\vee$ Dur(t_2) > 0)

That is, time elements with zero duration cannot meet each other.

A6. $\forall t_1,t_2 \in \boldsymbol{T}$ (Meets(t_1, t_2) $\Rightarrow$ Dur($t_1 \oplus t_2$) = Dur(t_1) + Dur(t_2))

That is, the "ordered union" operation,$\oplus$, over time elements is consistent with the conventional "addition" operation over the duration assignment function, i.e., 'Dur'.

N.B. In the time theory **T** introduced here, we adopt the following results of real number theory:

[1] The set of real numbers is totally ordered by the less-than-or-equal-to relation '$\leq$', where '>' is the 'bigger than" relation, that is, not($\leq$).

[2] '+' is the conventional addition operator over (non-negative) real numbers.

In terms of the 'Meets' relation, other exclusive order relations over time elements can be derived as below:

Equal(t_1, t_2) $\Leftrightarrow \exists t', t''$(Meets($t'$, t_1) $\wedge$ Meets(t', t_2) $\wedge$ Meets(t_1, t'') $\wedge$ Meets(t_2, t''))

Before(t_1, t_2) $\Leftrightarrow \exists t$(Meets($t_1$, t) $\wedge$ Meets(t, t_2))

Overlaps(t_1, t_2) $\Leftrightarrow \exists t,t_3,t_4(t_1 = t_3 \oplus t \wedge t_2 = t \oplus t_4)$

Starts(t_1, t_2) $\Leftrightarrow \exists t(t_2 = t_1 \oplus t)$

During(t_1, t_2) $\Leftrightarrow \exists t_3,t_4(t_2 = t_3 \oplus t_1 \oplus t_4)$

Finishes(t_1, t_2) $\Leftrightarrow \exists t(t_2 = t \oplus t_1)$

After(t_1, t_2) $\Leftrightarrow$ Before(t_2, t_1)

Overlapped-by(t_1, t_2) $\Leftrightarrow$ Overlaps(t_2, t_1)

Started-by(t_1, t_2) $\Leftrightarrow$ Starts(t_2, t_1)

Contains(t_1, t_2) $\Leftrightarrow$ During(t_2, t_1)

Finished-by(t_1, t_2) $\Leftrightarrow$ Finishes(t_2, t_1),

Met-by(t_1, t_2) $\Leftrightarrow$ Meets(t_2, t_1)

On one hand, the completeness of the 13 possible exclusive order relations (the above 12 plus Meets) between any two time elements can be simply characterized by a single axiom as below:

$$\forall t_1,t_2(Equal(t_1, t_2) \vee Before(t_1, t_2) \vee After(t_1, t_2) \vee Meets(t_1, t_2) \vee Met\text{-}by(t_1, t_2)$$

$$\vee Overlaps(t_1, t_2) \vee Overlapped\text{-}by(t_1, t_2) \vee Starts(t_1, t_2) \vee Started\text{-}by(t_1, t_2)$$

$$\vee During(t_1, t_2) \vee Contains(t_1, t_2) \vee Finishes(t_1, t_2) \vee Finished\text{-}by(t_1, t_2))$$

On the other hand, the exclusiveness of these 13 order relations needs to be characterized by the axioms of the following form[All1984]:

$$\forall t_1,t_2(\neg Relation1(t_1, t_2) \vee \neg Relation2(t_1, t_2))$$

where Relation1 and Relation2 are two distinct relations from the above 13 relations.

For the convenience of expression, we shall extend Allen’s non-exclusive relation ’In‘, which is defined for intervals alone [All1984], to accommodate both time intervals and points, and in addition, to introduce another temporal relation, ’Part‘, as below:

$$In(t_1, t_2) \Leftrightarrow Starts(t_1, t_2) \vee During(t_1, t_2) \vee Finishes(t_1, t_2)$$

$$Part(t_1, t_2) \Leftrightarrow Equal(t_1, t_2) \vee In(t_1, t_2)$$

Analogous to the conventional concepts and terminologies in real number theory, Then [Ma2007] summarized an ontological glossary with respect to time elements as below:

1) Time element t is a point if Dur(t) = 0.
2) Time element t is an interval if Dur(t) > 0.
3) Interval i is decomposable if there are two time elements t1 and t2 such that i = t1 ⊕ t2; otherwise, interval i is non-decomposable.
4) Interval i is a subinterval of interval j if Part(i, j).

   (For example, intervals [0,5], [0,5), (0,5], (0,5) and [1,4] are subintervals of interval

[0,5])

5) Interval i is a proper subinterval of interval j if In(i, j).

(For example, intervals [0,5), (0,5], (0,5) and [1,4] are proper subintervals of interval [0,5]; where [0,5] is not a proper subinterval of interval [0,5])

6) Interval i is a strict-proper subinterval of interval j if In(i, j) $\wedge$ Dur(i) < Dur(j).

(For example, intervals [1,4] is a strict-proper subinterval of interval [0,5]; where [0,5], [0,5), (0,5] and (0,5) are all not strict-proper subintervals of interval [0,5])

7) Point p is an internal point of interval i if In(p, i).

8) Point p is an inner point of interval i if During(p, i).

9) Point p is the left-ending point of interval i if Starts(p, i).

10) Point p is the right-ending point of interval i if Finishes(p, i).

11) Point p is the left-bounding point of interval i if Starts(p, i) $\vee$Meets(p, i).

12) Point p is the right-bounding point of interval i if Finishes(p, i) $\vee$Met-by(i, p).

13) Interval i is left-open at point p if Meets(p, i).

14) Interval i is right-open at point p if Meets(i, p).

15) Interval i is left-closed at point p if Starts(p, i).

16) Interval i is right-closed at point p if Finishes(p, i).

By definition, an internal point of a given interval is either the left-ending point or the right-ending point, or an inner point of the interval. In addition, an interval is left-closed at a point if and only if the point is the left-ending point of that interval. Similarly, an interval is right-closed at a point if and only if the point is the right-ending point of that interval [Ma2007].

It is important to note that, while a left-ending point of an interval is also the left-bounding point of that interval; a left-bounding point is not necessarily the left-ending point. Similarly, while a right-ending point of an interval is also the right-bounding point of that interval; a right-bounding point is not necessarily the right-ending point.

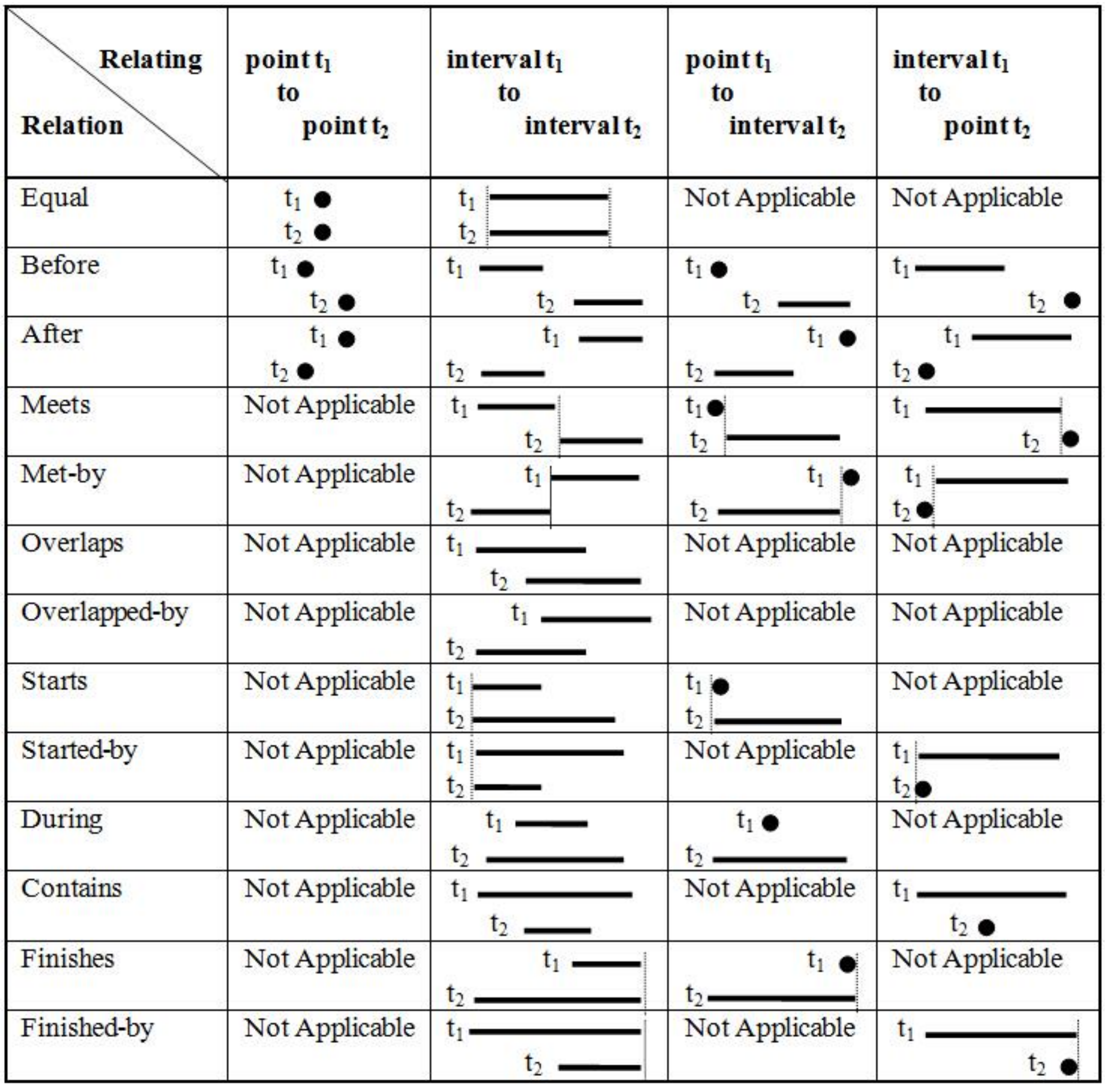
In addition, although there are no definitions about the ending points for intervals, [MK2003] introduced the expression of the "open" and "closed" nature of intervals, which can be formally defined as:

- Interval *i* is *left-open* if and only if there is a point *p* such that Meets*(p, i)*;
- Interval *i* is *right-open* if and only if there is a point *p* such that Meets*(i, p)*;
- Interval *i* is left-closed if and only if there is a point *p* and an interval *i'* such that Meets*(i', i)* $\wedge$ Meets*(i', p)*;
- Interval *i* is *right-closed* if and only if there is a point *p* and an interval *i'* such that Meets*(i, i')* $\wedge$ Meets*(p, i')*.

The above interpretation about the "open" and "closed" nature of primitive intervals is in fact consistent with the conventional meaning of the open and closed nature of intervals of real numbers. For instance, interval (2, 5] which does not include number 2 itself is "left-open", since (point) 2 is an immediate predecessor of interval (2, 5]. Similarly, (2, 5] which does include number 5 is "right-closed", since both point 5 and interval (2, 5] are immediate predecessors of interval (5, 8). But point 5 is not an immediate predecessor of interval [5, 8] or [5, 8) according to the above interpretation.

Based on Allen and Hayes's thirteen temporal relations between intervals only, **PIBS** extended them to 30 possible temporal relations for both point and interval (See Table 2.2):

Table 2.2 Four Groups of Time Relations



Relating / Relation	point t_1 to point t_2	interval t_1 to interval t_2	point t_1 to interval t_2	interval t_1 to point t_2
Equal	t_1 t_2	t_1 t_2	Not Applicable	Not Applicable
Before	t_1 t_2	t_1 t_2	t_1 t_2	t_1 t_2
After	t_1 t_2	t_1 t_2	t_1 t_2	t_1 t_2
Meets	Not Applicable	t_1 t_2	t_1 t_2	t_1 t_2
Met-by	Not Applicable	t_1 t_2	t_1 t_2	t_1 t_2
Overlaps	Not Applicable	t_1 t_2	Not Applicable	Not Applicable
Overlapped-by	Not Applicable	t_1 t_2	Not Applicable	Not Applicable
Starts	Not Applicable	t_1 t_2	t_1 t_2	Not Applicable
Started-by	Not Applicable	t_1 t_2	Not Applicable	t_1 t_2
During	Not Applicable	t_1 t_2	t_1 t_2	Not Applicable
Contains	Not Applicable	t_1 t_2	Not Applicable	t_1 t_2
Finishes	Not Applicable	t_1 t_2	t_1 t_2	Not Applicable
Finished-by	Not Applicable	t_1 t_2	Not Applicable	t_1 t_2

- Relations relating a point to a point:

  {Equal, Before, After}

- Relations relating a point to an interval:

{Before, Meets, Starts, During, Finishes, Met-by, After}

- Relations relating an interval to a point:

  {Before, Meets, Started-by, Contains, Finished-by, Met-by, After}

- Relations relating an interval to an interval:

  {Equal, Before, Meets, Overlaps, Starts, During, Finishes, Finished-by, Contains, Started-by, Overlapped-by, Met-by, After}

Then [MK1994] argues that the axiomatisation can be interpreted in various temporal models: dense or discrete, linear or branching, etc.

Section 2.1.4 Uncertainty And Incompleteness In Temporal Logic

In the domain of Artificial Intelligence, temporal knowledge can be uncertain and incomplete due to various reasons:

- Temporal knowledge may come from, or go into, more than one possible world (e.g., after arriving at Venice in the morning, the visitor may take a train in the afternoon, or take a flight in the evening, to get to Rome);

- Temporal knowledge may be only relative (e.g., "during the time when the officer was in his office", "after 9 o'clock", etc., which refer to times that are known only by their relative temporal relations to other temporal reference), rather than being absolute (e.g., "8 pm on the 8th of August 2008", "the last week of August 2008", which refer to times with absolute values);

- Temporal duration may be only relative (e.g., "less than 6 hours", "more than 12 years but less than 15 years", etc., which refer to some uncertain amount of temporal granularity), rather than being absolute (e.g., “31 minutes”, “18 hours”, etc., which refer to some certain amount of temporal granularity);

- One may only know event E_A occurred “Before” event E_B, without knowing their precise starting and finishing times, what happened between E_A and E_B, or how long the delay was between E_A and E_B.

1. Incompleteness in temporal logic

Researchers must carefully distinguish between an ontological representation of temporal relations, i.e., the representation of a specific set of mutually compatible temporal relations, and the representation of knowledge about temporal relations [CTR1992]. If all the relations of events are precise, the distinction is insignificant; but if there is an incomplete knowledge, it will be very different. Typically, we do not have complete knowledge about temporal relations between events to start with; but even if we do, after only one inference step we may not have complete knowledge about the inferred relations.

For example, one may only have temporal information about the birth or the death of an individual, but not both; or one may know that a certain event Y did not start before a given event X, but we do not know if X and Y started simultaneously or if Y started after X. For another example, in order to determine that the First World War was before the Second World War, it is sufficient to know that the First World War's end time took place before the Second World War's start time; it does not help if in addition we know when the First World War was started or when the Second World War ended. In many cases useful inferences can be drawn from such incomplete knowledge, in some cases even without any loss of information.

However, Point and Interval based system is strong enough to express the incompleteness of a given situation. [CTR1992] gives an example with Allen's representation. The situation *"X was born before Y's death"* can be expressed as "X lived before Y or X's life meets Y's life or X's life overlaps Y's life or X's life is finished by Y's life or X's life contains Y's life or X's life is started by Y's life or X's life equals Y's life or X's life starts Y's life or X lived during Y's life or X finishes Y's life or X's life is overlapped by Y's life". The situation "X died after Y" can be expressed as "X' life contains Y's life or X' life is started by Y's life or X' life is overlapped by Y's life or X' life is started by Y's life or X lived after Y". The inference step then consists of forming the conjunction of the two sets of disjunctions: "X' life contains Y's life or X' life is started by Y's life or X' life is overlapped by Y's life" which is equivalent to the conclusion derived above [CTR1992].

It has been noted that the absolute-time-stamping of temporal data provides an efficient indexing method for temporal systems, but suffers from the requirement that precise time values for all temporal data need to be available.

2. Uncertainty about temporal relations

One may not know if event X takes place before event Y, if X meets Y, or if X overlaps Y. However, he/she can distinguish these three options from the remaining ten temporal relations.

On the other hand, sometimes information about a historical event can only be deduced from documents reporting about related events. Often several documents state contradictory facts about some event. As an example consider Stalin's birth date. Officially, for the USSR it is 21 December 1879 but according to church records his birth was registered as 6 December 1878. Historians are in disagreement over which time specification is the right one. In such a case, we say that the temporal specification of the event is uncertain. In natural language, such uncertain temporal specifications are often written informally as? - 1640 (meaning that the beginning time of the event is unknown) or as ca. 1801 (meaning circa, or around this year) [GB2003].

Kim [KH1995] identifies four different types of uncertainty:

- Observation uncertainty: uncertainty about the values of observations returned by sensors or other devices.
- Non-deterministic action laws: effects of actions can be non-deterministic, leading to uncertainty about sequences of an action.
- Persistence duration uncertainty: reasoning correctly about some properties require to know how long they persist, but their period of persistence cannot be stated precisely in many cases.
- Action time uncertainty: we may not know precisely when an action occurred. This uncertainty can affect both the hanging time of properties and the properties themselves.

Section 2.2 Natural Language Process

Generally speaking, natural information includes voice, written words (or signs) and body language. The key of NLP is to make the computer understand natural language and output the corresponding result humans need. The NLP glossary has been listed in Appendix E.

Training data generally takes a lot of work to create, then a pre-existing corpus is typically used. These usually use the **Penn Treebank** [MSM1993] or **Brown Corpus** [FK1983] tags. The most common part of speech (POS) tag schemes are those developed for the Penn Treebank and Brown Corpus. Penn Treebank is probably the most common, but both corpora are available with NLTK. This thesis will use Penn Treebank POS Tags. Table 2.3 shows parts of Penn Treebank POS Tags which will be used later.

Table 2.3 Speech of Tag

POS Tag	**Description**	**Example**
VB	verb, base form	take
VBD	verb, past tense	took
VBG	verb, gerund/present participle	taking
VBN	verb, past participle	taken
VBP	verb, sing. present, non-3d	take
VBZ	verb, 3rd person sing. present	takes

For Temporal Natural Language Process (TNLP), all the actions and times need to be extracted out of natural knowledge. As has been discussed in [AF1994], for any temporal knowledge, the world really does not contain events. For instance, consider a circumstance in which a ball rolled off a table and bounced on the floor. However, this description can also be described as the ball dropped. No one description is more correct than the other. Allen thought that it was quite meaningless to discuss events although some may be more informative for certain circumstances. Instead, action is more appropriate to describe natural knowledge. An action may cause an event to occur, which in turn may cause other desired events to also occur.

Figure 2.2 shows us that for any given event or action, it can always be decomposed into several elements. It is quite clear both action and event are not the basic elements. It totally depends on the problem to be solved.

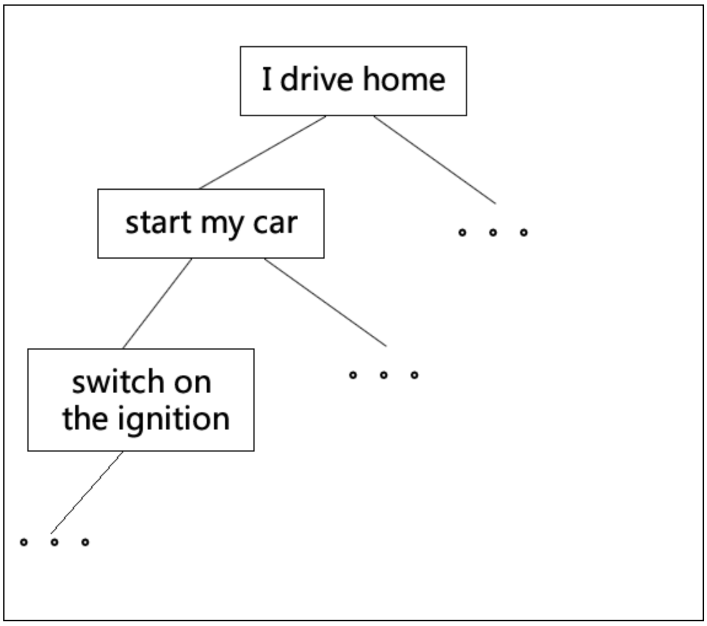


Figure 2.2 Event Or Action

This thesis gives the following explanations on event and action:

1. One event can consist of one or more actions. For example, "I drove home" includes "I started the car", "I drove on the road", " I got home" and so on;

2. One action may also activate a new event. For example, Action "I got home" may also be performed in the hope of causing the event "I walked my dog".

3. One action can be included in different events. For example, "2:00PM today" can be both *during* "this month" and "this year".

Section 2.2.1 Natural Language Process

NLP is related to the area of human–computer interaction. Many challenges in NLP involve natural language understanding, that is, enabling computers to derive meaning from human or natural language input, and others involve natural language generation. However, the most commonly researched tasks in NLP are Text categorization, Information retrieval, Information extraction, Text-proofing, Question answering and so on.

According to recent works, modern NLP algorithms are based on machine learning, especially statistical machine learning. Compared to most of the previous works, machine learning often learn rules through the analysis of a real-world example, which has been hand-annotated with the correct values to be learned. The main basic approaches and algorithms of machine learning can be summarized as follows:

1. **Decision tree learning:** Decision tree learning uses a decision tree as a predictive model, which maps observations about an item to conclusions about the item's target value [RLM2008].
2. **Association rule learning:** Association rule learning is a method for discovering interesting relations between variables in large databases [AIS1993].

3. **Artificial Neural Networks:** An artificial neural network (ANN) learning algorithm, usually called "neural network" (NN), is a learning algorithm that is inspired by the structure and functional aspects of biological neural networks [PPE1999].

4. **Inductive Logic Programming:** Inductive logic programming (ILP) is a sub-field of machine learning which uses logic programming as a uniform representation for examples, background knowledge and hypotheses [SE1991].

5. **Genetic Algorithms:** A genetic algorithm (GA) is a search heuristic that mimics the process of natural selection, and uses methods such as mutation and crossover to generate new genotype in the hope of finding good solutions to a given problem [GDH1998] [MST1994].

Section 2.2.2 Stanford NLP

The Natural Language Processing Group at Stanford University is very talented at making computers processing and understanding human languages. Their work ranges from basic research in computational linguistics to key applications in human language technology, and covers areas such as sentence understanding[SKL2014], machine translation [ZSC2013], probabilistic parsing [MC2000] and tagging [MAN2011], bio-medical information extraction [FDM2005], grammar induction [SAJ2013], word sense disambiguation [KTI2002], and automatic question answering [WM2010].

There are many famous NLP toolkits such as TimeML [BK2005], NLTK toolkit[LB2002] , SUTime [CM2012] etc. The reason why this thesis chooses SUTime is that both TimeML and NLTK toolkit has some problem called *self-referring problem* on information extraction (See Section 2.3.1). The aim of Visual Time is to support both Chinese and English. This section will introduce two NLP tools **Stanford Parser** and **Stanford Temporal Tagger** (SUTime) [CM2012]. The version used in this thesis is ***stanford-corenlp- full- 2014-01-04***.

1. Stanford Parser

Stanford Parser is a multilingual tool, which supports English, Chinese, German and Arabic [RJCA2013] [DKC2003] [MBC2006] [AC2008] [RC2003] [PHD2009] [SC2010] [SMJ2011]. The parser has also been used in other languages, such as Italian, Bulgarian, and Portuguese. This thesis will discuss the English parser [DC2003].

A probabilistic context free grammars (PCFG) structure was used as follows (See Figure 2.3):

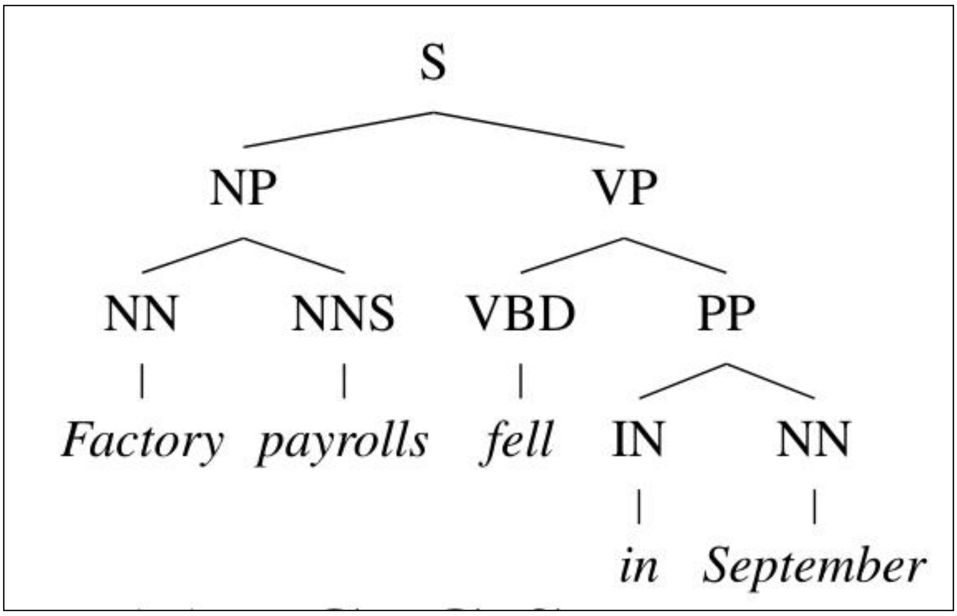


Figure 2.3 PCFG Structure

Shortcomings: However, Stanford Parser can perfectly extract all the VB, VBG, VBZ, VBD, VBN and VBP from natural knowledge. However, if one wants to get all the action of natural knowledge, some improved algorithms are required. For example,

Figure 2.4 (a) shows its effect on the sentence "I start to work". If one wants to extract the action VB "work", then VBD "start" needs to be ignored when extracting. In sentence "I started to work when Tom left.", VBD "started" cannot be ignored because it reflects the temporal relation "*starts*" between VB "work" and VBD "left" (See Figure 2.4).

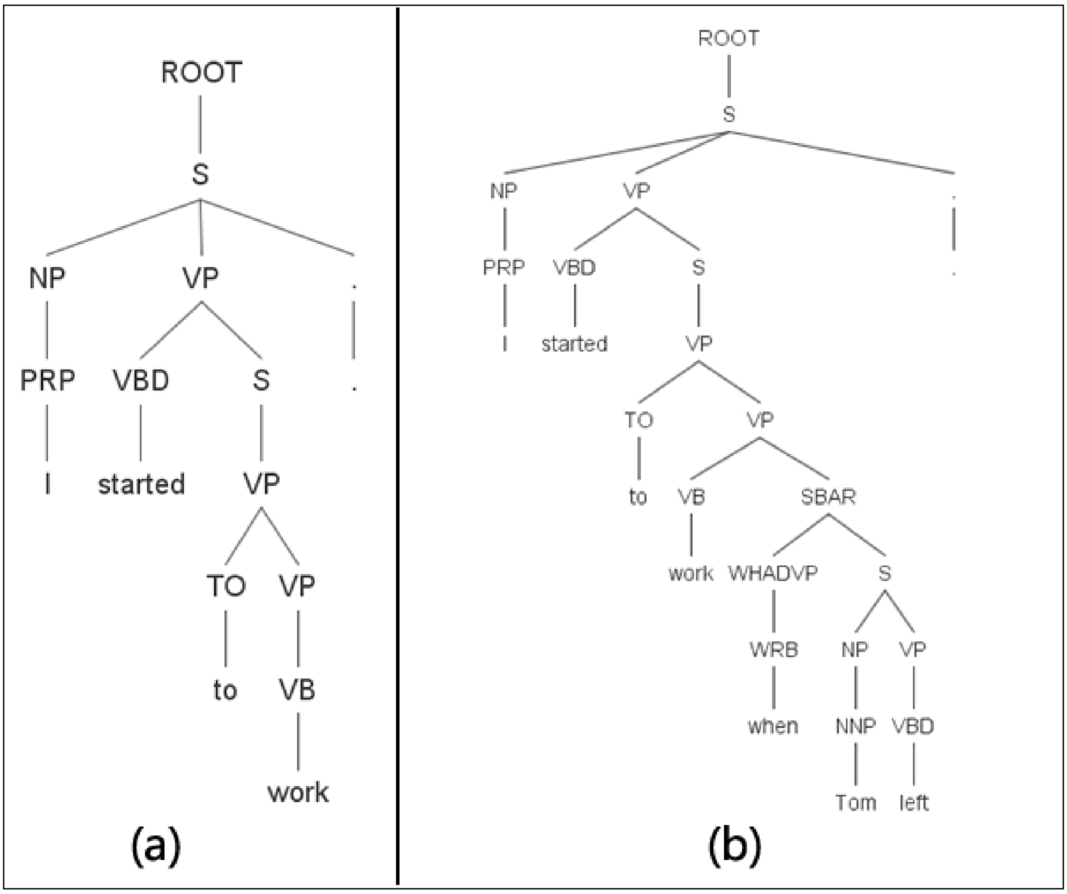


Figure 2.4 First Shortcoming of Stanford Parser

2. Stanford Temporal Tagger (SUTime)

SUTime is a very useful tool for extracting temporal information with a format like 3:00PM, 3pm and so on [CM2012]. Table 2.4 shows the expression format of Time (with **2011-09-19** as the reference date).

Table 2.4 Expression Format

Expression	*Type*	*Value*
October of 1963	DATE	1963-10
October	DATE	2011-10
last Friday	DATE	2011-09-16
next weekend	DATE	2011-W39-WE
the day after tomorrow	DATE	2011-09-21
the nineties	DATE	199X
winter of 2000	DATE	2000-WI
5th century B.C.	DATE	-05XX
now	DATE	PRESENT_REF
Saturday morning	TIME	2011-09-24TMO
4 p.m. Tuesday	TIME	2011-09-20T16:00

Duration: The amount of intervening time between the two end-points of a time interval. Durations can be specified as a combination of a unit (e.g., day, month, year, etc.) and a numeric value (the quantity associated with the unit). SU-TIME recognizes three types of durations:

- **Exact durations**: Both value and unit fully specified;
- **Inexact durations**: Unit known, but not its value; and
- **Duration ranges**: The duration is bounded between a minimum and a maximum duration.

Examples of temporal expressions corresponding to the three different types of durations are given in Table 2.5:

Table 2.5 Temporal Expressions

	Expression	*Type*	*Value*
Exact	3 days	DURATION	P3D
Inexact	a few years	DURATION	PXY
Range	2 to 3 months	DURATION	P2M/P3M

SUTIME applies three types of rules, in the following order:

1. **Text regex rules**: mappings from simple regular expressions over characters or tokens to temporal representations;

2. **Compositional rules**: mappings from regular expressions over chunks (both tokens and temporal objects) to temporal representations; compositional rules are iteratively applied after the text regex rules. At each stage, nested time expressions are removed, and these rules are applied until the final list of time expressions stabilizes;

3. **Filtering rules**: the final stage, in which ambiguous expressions that are likely not to be temporal expressions are removed from the list of candidates. For instance, we can specify a rule indicating that if a potential temporal expression is a single word, fall, and the part of speech tag is not a noun, then it is likely that fall refers to the act of falling and not the season autumn, and so SUTIME will refrain from marking it as a temporal expression.

An Example:

I got home before 3:00 pm.

Figure 2.5 shows the corresponding result.

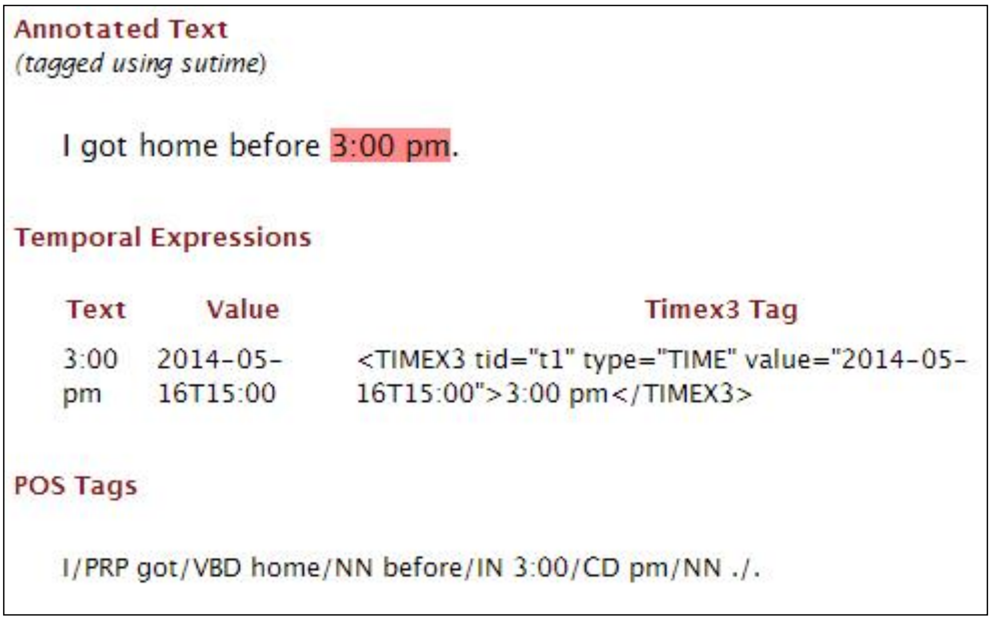


Figure 2.5 SUTime Example

Shortcoming: Using Stanford parser and Stanford Temporal Tagger (SUTime), it is not hard to pick out all the verbs and time tags from the natural language information. But for temporal modelling, it is not enough, for example:

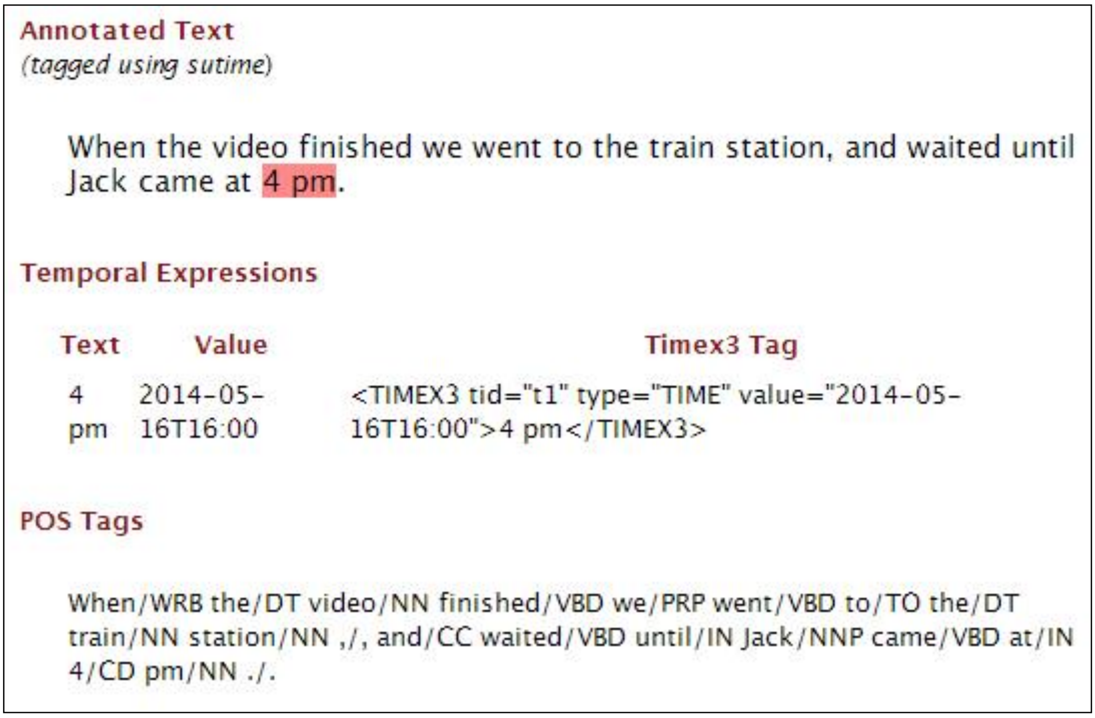
When the video finished we went to the train station,

and waited until Jack came at 4 pm.

Figure 2.6 shows the result of above sentence when using SUTime. But the POS Tags of its result still simply splits natural information into four verbs (*finished/VBD, went/VBD, waited/VBD, came/VBD*). *Finished/VBD* reflects the start time of action *went/VBD* and end time of action "the video is on". This means we can get:

<HH>Meets (video_on, go_to_train_station)

It is quite clear that an improved algorithm should be designed to get the results like <HH>.



Annotated Text
(tagged using sutime)

When the video finished we went to the train station, and waited until Jack came at 4 pm.

Temporal Expressions

Text	Value	Timex3 Tag
4 pm	2014-05-16T16:00	<TIMEX3 tid="t1" type="TIME" value="2014-05-16T16:00">4 pm</TIMEX3>

POS Tags

When/WRB the/DT video/NN finished/VBD we/PRP went/VBD to/TO the/DT train/NN station/NN ,/, and/CC waited/VBD until/IN Jack/NNP came/VBD at/IN 4/CD pm/NN ./.

Figure 2.6 SUTime Shortcoming

There is no doubt that SUTime and Stanford Parser are quite good at extracting times and verbs of natural knowledge, but an improved algorithm is required to solve their shortcoming.

Section 2.3 Program Visualization

[MYE1990] gives the definition of “Programming”, “Visual Programming” and “Program Visualization”.

Programming: a computer ‘‘program’’ is defined as “a set of statements that can be submitted as a unit to some computer system and used to direct the behaviour of that system [DW1983]”. While the ability to compute ‘‘everything’’ is not required, the system must include the ability to handle variables, conditionals and iteration, at least implicitly [MYE1990].

Visual Programming (VP): VP refers to any system that allows the user to specify a program in a two (or more) dimensional fashion. Although this is a very broad definition, conventional textual languages are not considered two dimensional since the compilers or interpreters process them as long, one-dimensional streams [MYE1990]. Visual Programming does not include systems that use conventional (linear) programming languages to define pictures, such as, Sketchpad [Sut1963], CORE, PHIGS, Postscript [Ado1985], the Macintosh Toolbox [App1985], or X-11 Window Manager Toolkit [MA1988], since these do not create ‘‘programs’’ as defined above [MYE1990].

Program Visualization (PV): PV is an entirely different concept from Visual Programming. In Visual Programming, the graphics is used to create the program itself, but in Program Visualization, the program is specified in a conventional, textual manner, and the graphics is used to illustrate some aspect of the program or its run-time execution. Unfortunately, in the past, many Program Visualization systems have been incorrectly labelled [MYE1990].

This thesis introduces a visualized framework for uncertain and incomplete temporal knowledge which relates to the research field – Program Visualization. As Paolo Bottoni argues in [BLP1998], interaction *visual* is very meaningful because the image produced by the computer is the only access for the user to the state of the underlying computation. The interaction is completely visual if all the relevant aspects of the state are fully reflected in the produced image through some suitable characteristic structure.

[BCLM1995][BCLM1997][BLCM1998] claim that for each message produced either by the computer or the human, two interpretations exist: the first is the one intended by the generator and the second is associated by the receiver. These two interpretations are formalized as two functions μH and μC mapping an icon into its meaning for the human and the computer respectively. A *correct communication* occurs if a pair of inverse morphisms *h* and *c* can be established between the two interpretations, as shown in Fig.2.7:

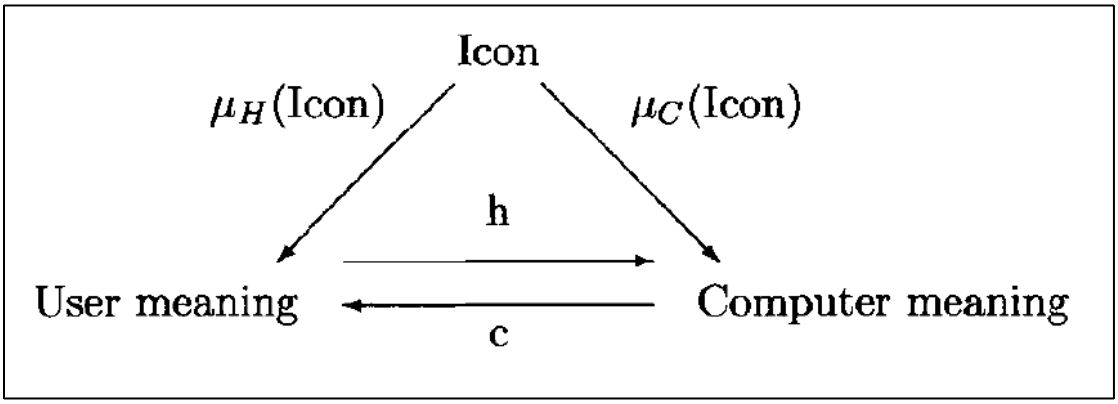


Figure 2.7 Morphisms between Icons and Meanings

[Mye1990] introduces three applications of Program Visualization:

- helping to teach algorithms involving data structures;
- helping to teach program concepts;

- helping to debug programs.

Section 2.4 A Visualized Temporal Framework

[EB2010] contributes on TimeML and the temporal representing in Turkish. [EB2010] points out that both TimeML and Allen's Interval Logic are not sufficient for a representation of Turkish temporal logic.

Firstly, one of the specific problems for Turkish temporal logic is the positive/negative verb repetition. For example, in English a single verb like "blink" is represented in Turkish with two separate verbs "yanıpsönmek" (to light and to fade). This concept can also be represented by "to flash" or "to twinkle" in English [EB2010]. The examples above show a term in Turkish while ATL is insufficient. Because the precise order of words in Turkish temporal logic is not important.

Secondly, TimeML cannot represent a *self-referring event*.

For example, in the sentence "My current talk is about computer science", the speaker refers to the current talk (See Figure 2.8), which is the talk itself. This case creates a self-reference, which is not covered in Allen's interval logic, and therefore not in TimeML, either [Sad2012].

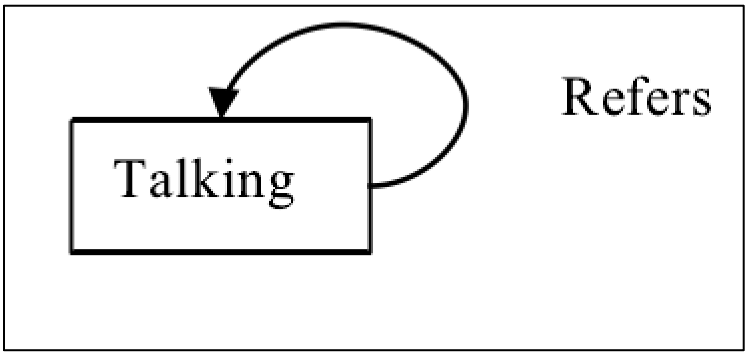


Figure 2.8 Self-referring Event

[Sad2012] puts forward an extended temporal relation (See Figure 2.9).

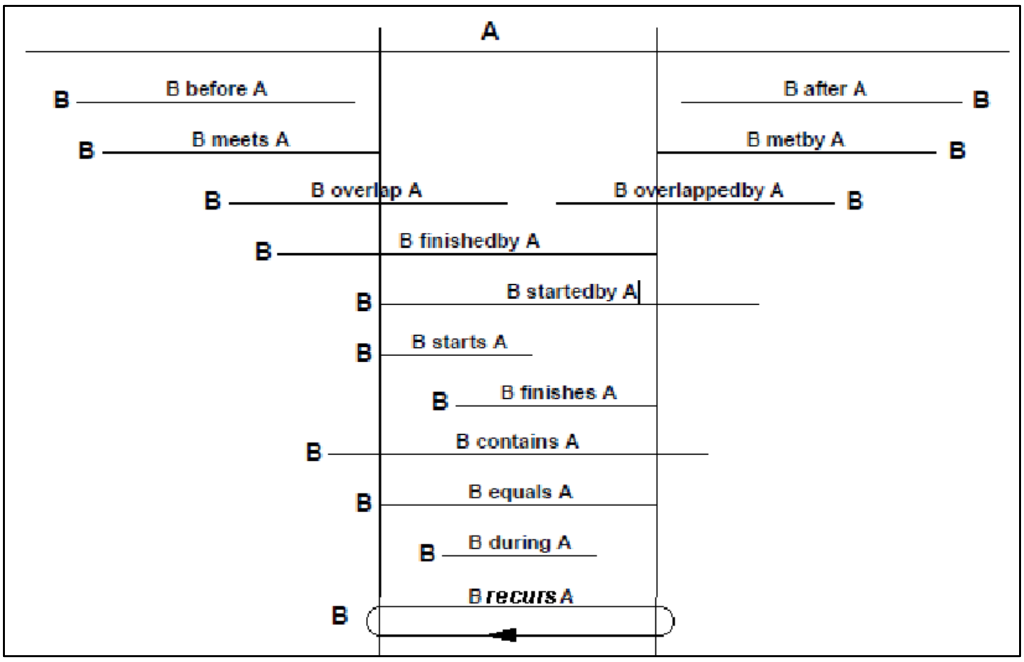


Figure 2.9 Extended Temporal Relations

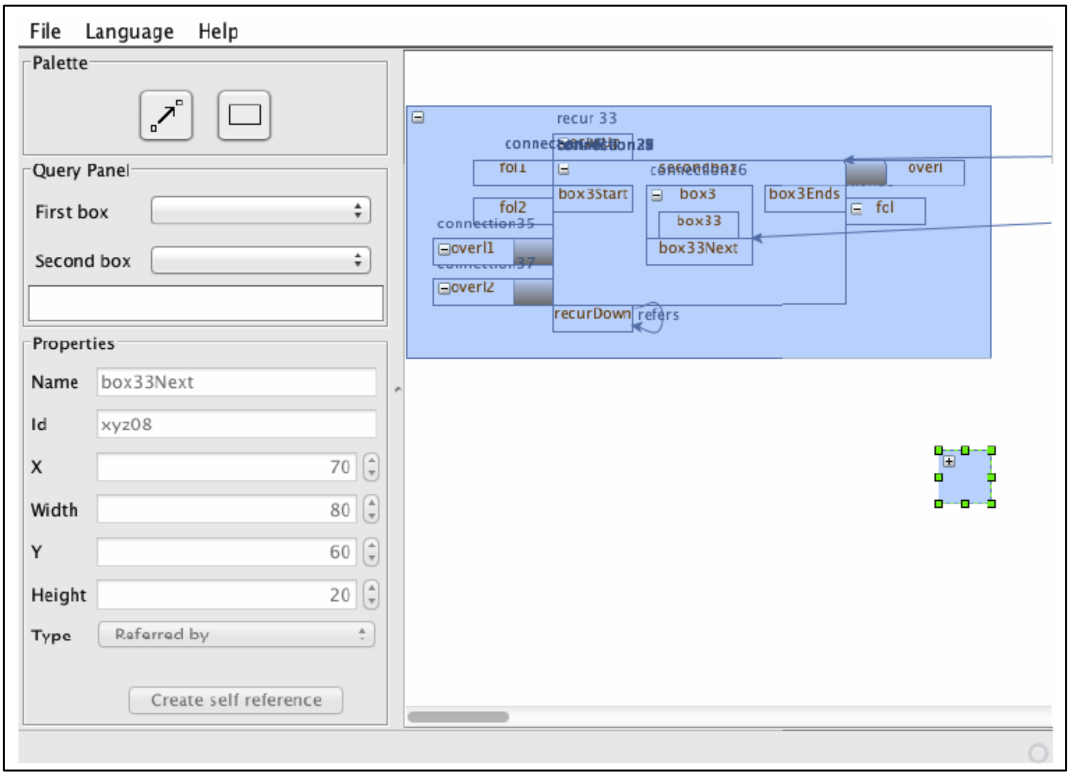


Figure 2.10 Novel Temporal Framework

Based on [EB2010] [Sad2012], Sadie put forward a visualized temporal framework (See Figure 2.10).

In Figure 2.11, Sadie visualizes the relation between the events “Attending classes” and “Take notes,” where the event “Take notes” is negative. However, Allen’s interval logic is strong enough to express the negative, which will be introduced in Chapter 3.

However, in our daily life, most of the information people collect from internet is uncertain or incomplete. One problem of this visual framework is that uncertain and incomplete temporal knowledge (UITK) is not supported.

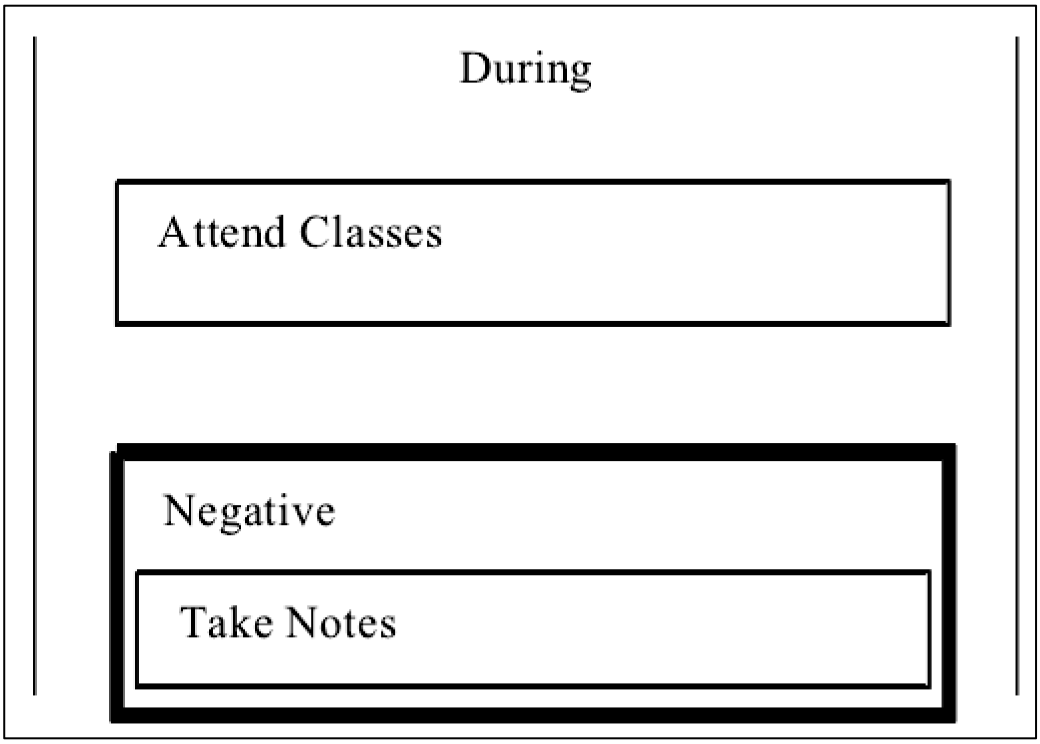


Figure 2.11 Relation of A Negative Event

Section 2.5 Conclusion of Literature Review

This thesis introduces a visualized framework for uncertain or incomplete temporal knowledge called Visual Time. The flow chart of Visual Time is shown in Figure 2.12. It is quite clear that Temporal Logic theory, NLP, Program Visualization are related research fields.

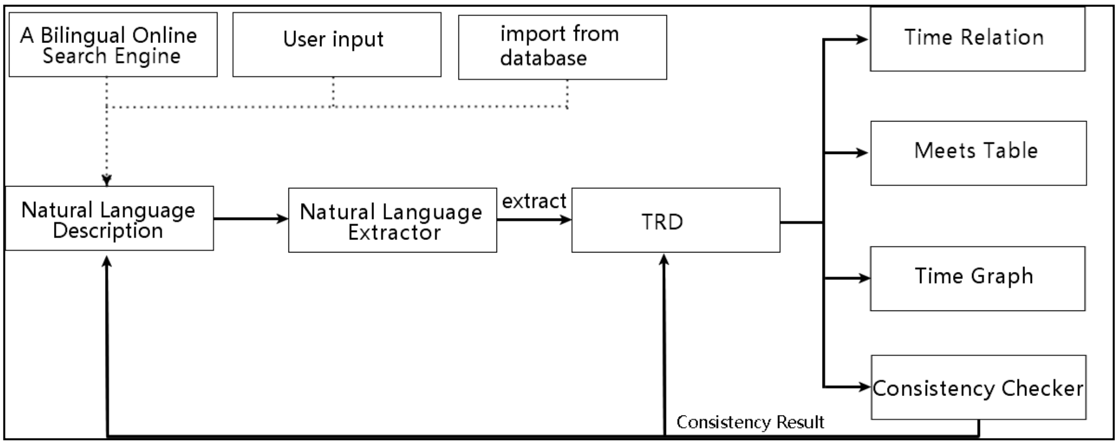


Figure 2.12 Flow Chart of Visual Time

Point and interval based system helped the research of temporal logic. The research on temporal basis helps this thesis to build the theoretical background of TRD.

Stanford NLP Parser and SUTime are used to filter verbs and temporal information from natural information. The filtered information will be improved and modelled by TRD.

Visual Time is related to the Program Visualization, this thesis puts forward a graphical temporal diagram called Time Relation Diagram (TRD) for representing the time relations among time elements which could be both point and interval.

CHAPTER 3 REPRESENTING UNCERTAIN AND INCOMPLETE TEMPORAL KNOWLEDGE

Base on the point and interval based time theory introduced in Chapter 2, as an extension of the work of [JBM2010], a graphical representation of uncertain and incomplete temporal knowledge is introduced in what follows in this chapter. Such a graphical representation expresses temporal knowledge in terms of a partially weighted directed graph, with the following semantic interpretation:

S1. Each time element is expressed as an arrowed-edge with a beginning-node and an ending node; and for time elements with known duration, the corresponding edges are weighted by their durations respectively (See Figure 3.1):

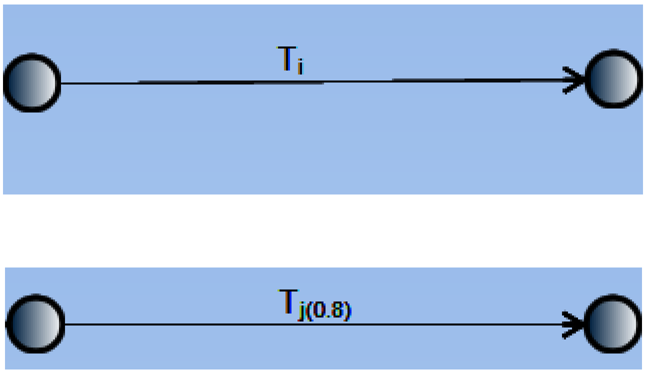


Figure 3.1 Time Elements

S2. The relation Meets(T_i, T_j) is presented by means of merging the target-note of time element T_i and the source-node of time element T_j. In other words, Meets(T_i, T_j) is denoted by the node structure where T_i is an in-edge and T_j is an out-edge of a same node, respectively (See Figure 3.2):

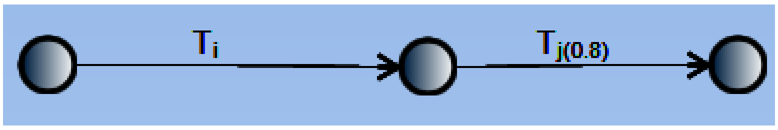


Figure 3.2 Meets(t_i, t_j)

S3. Meets(T_i, T_j) $\wedge$ Meets(T_i, T_k) is denoted by defining T_i as an in-arc and T_j and T_k as two out-arcs of the same node, respectively (See Figure 3.3):

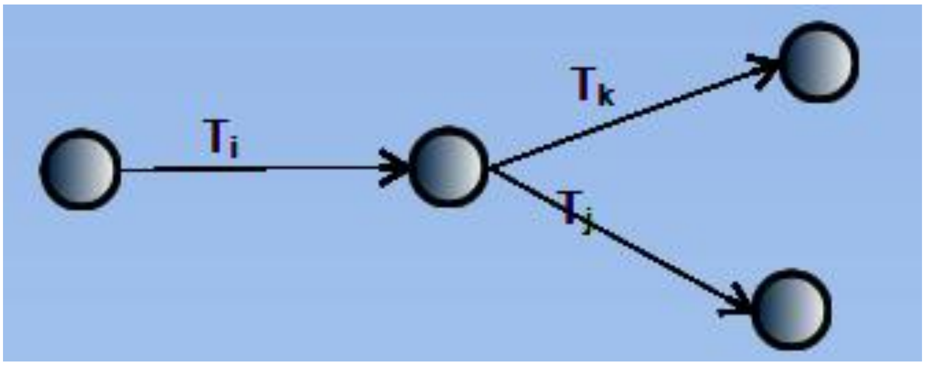


Figure 3.3 Meets(T_i, T_j) $\wedge$ Meets(T_i, T_k)

S4. Meets(T_i, T_k) $\wedge$ Meets(T_j, T_k) is denoted by defining T_i and T_j as two in-arcs and T_k as an out-arc of the same node, respectively (See Figure 3.4):

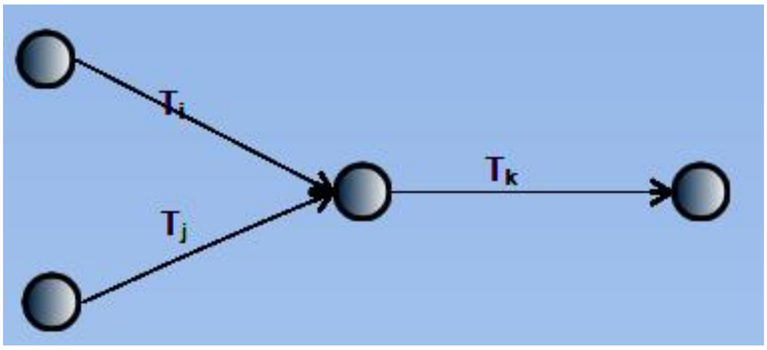


Figure 3.4 Meets(T_i, T_k) ∧ Meets(T_j, T_k)

Section 3.1 Graphically Representing Temporal Relations

The 13 temporal relations (Meets, Equal, Before, Overlaps, Starts, During, Finishes, Met-by, After , Overlapped-by, Started-by, Contains , Finished-by) over time elements can be expressed respectively from Figure 3.5 to Figure 3.17. As has been explained in Section 2.1.3, all the temporal relations can be characterized by the 'Meets' relation.

N.B. In table 2.2, it shows quite a lot of temporal relations that are not applicable for some time elements such as "*point cannot Meets point* "and so on.

The semantic interpretation of the graphical representation of the Meets relation can be given as below:

Relation Meets(T_1, T_2) is graphically presented by means of merging the target-node of time element T_1 and the source-node of time element T_2 as the same node, of which T_1 is an in-edge and T_2 is an out-edge, respectively (See Figure 3.5):

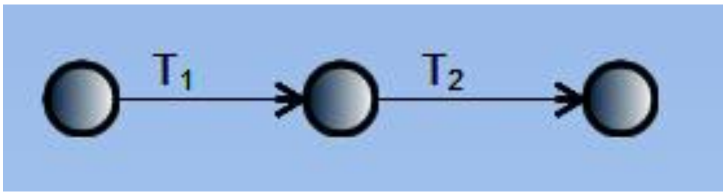


Figure 3.5 Meets (T_1, T_2)

In turn, other temporal relations can be graphically expressed as below:

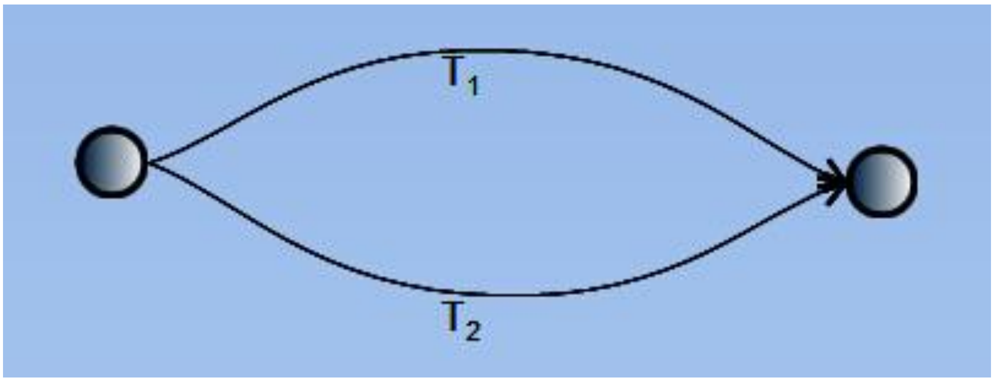


Figure 3.6 Equal (T_1, T_2)

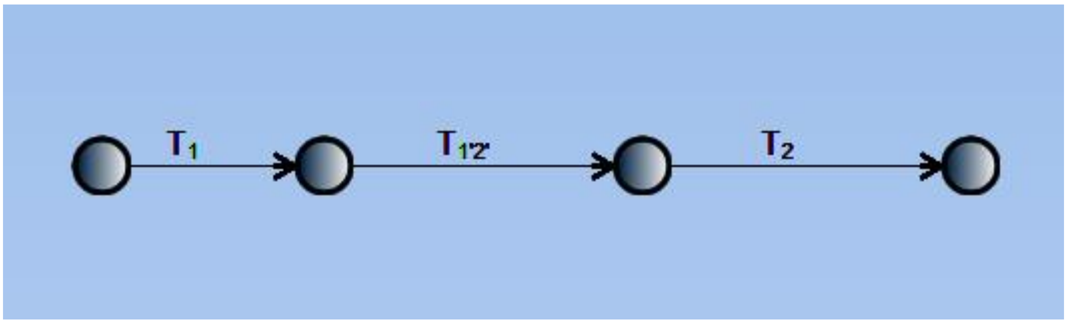


Figure 3.7 Before (T_1, T_2)

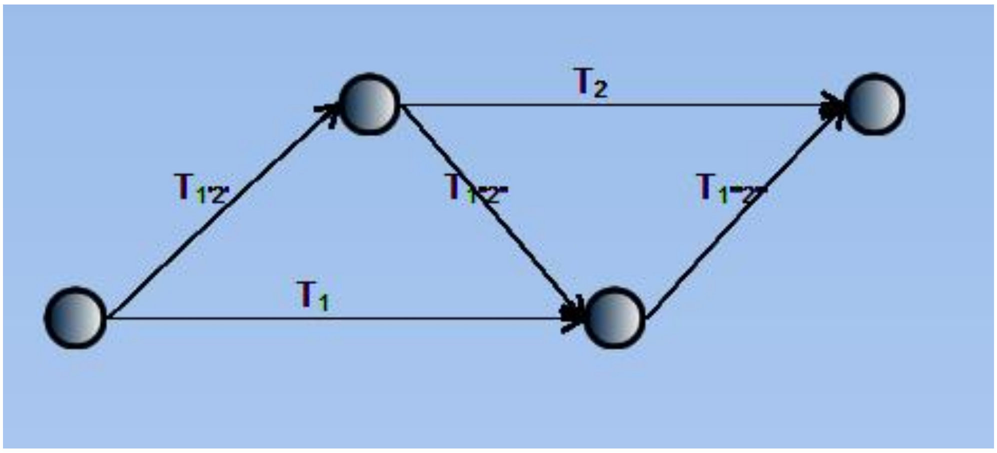


Figure 3.8 Overlaps (T_1, T_2)

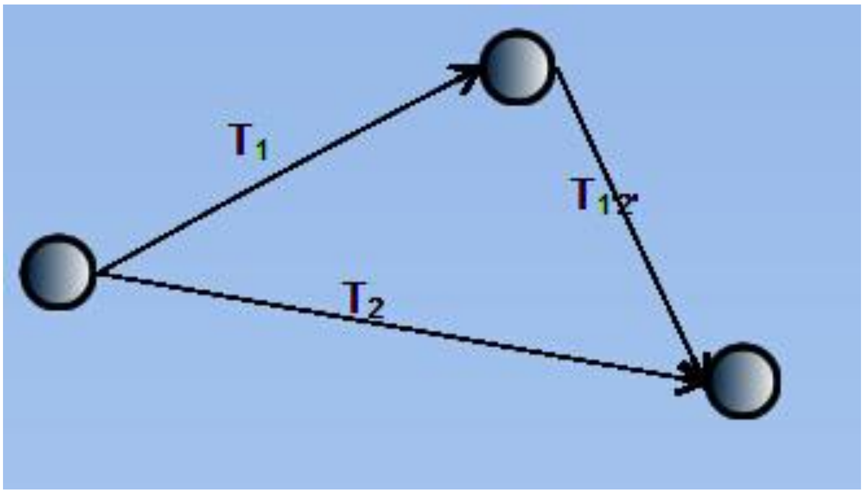


Figure 3.9 Starts (T_1, T_2)

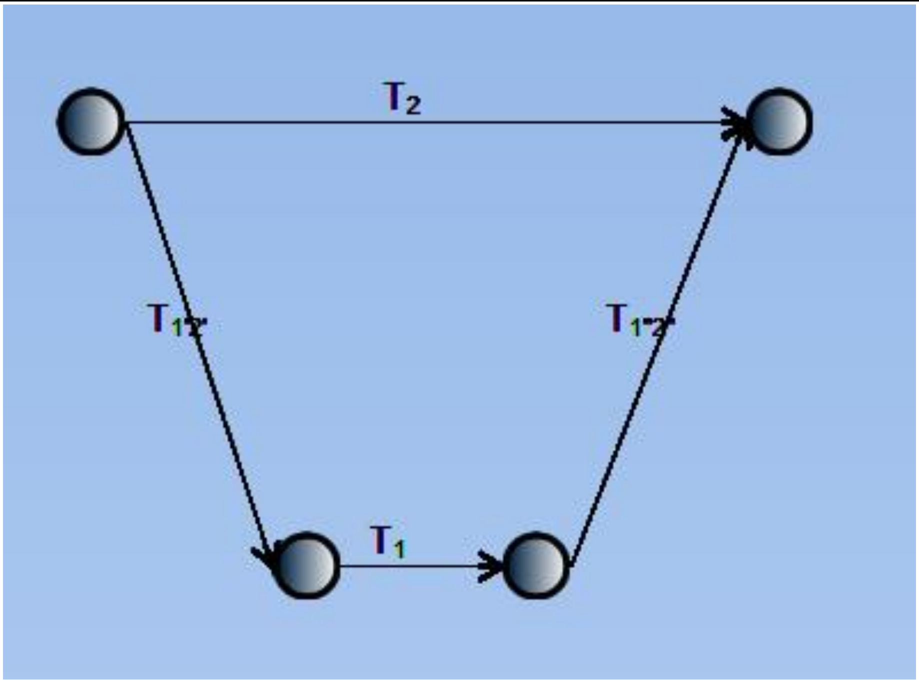


Figure 3.10 During (T_1, T_2)

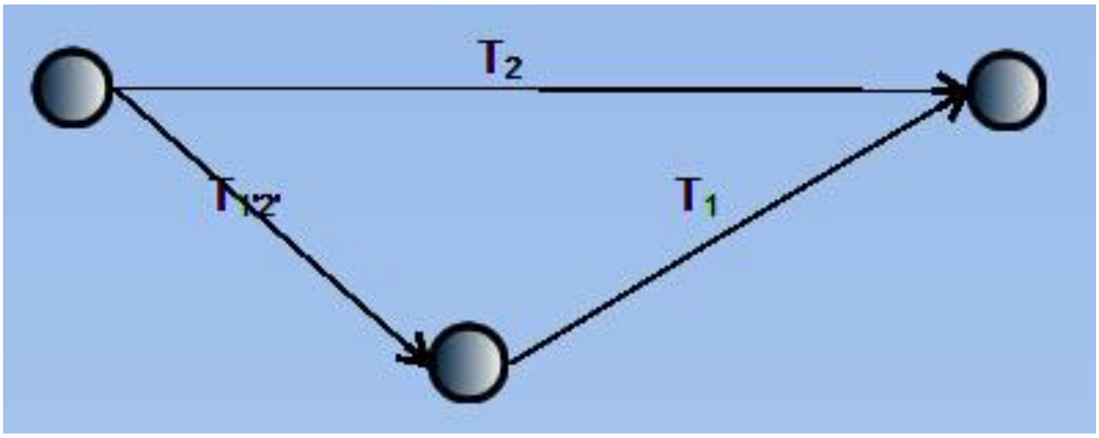


Figure 3.11 Finishes (T_1, T_2)

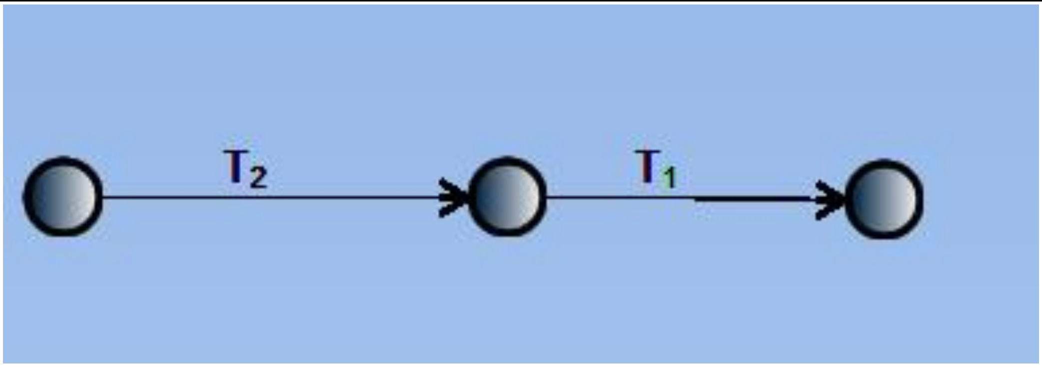


Figure 3.12 Met-by(T_1, T_2)

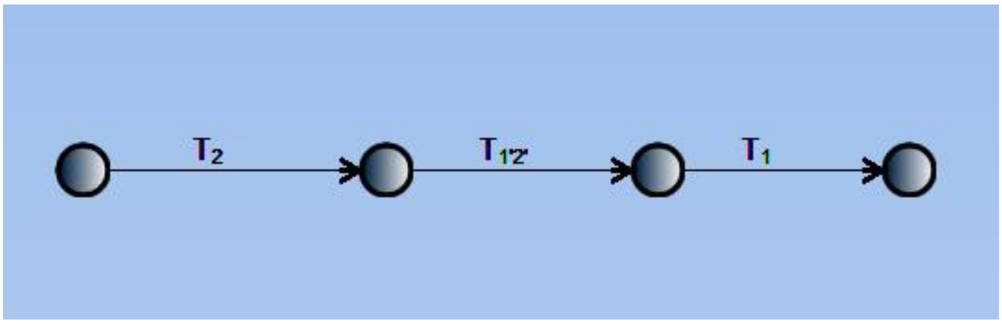


Figure 3.13 After (T_1, T_2)

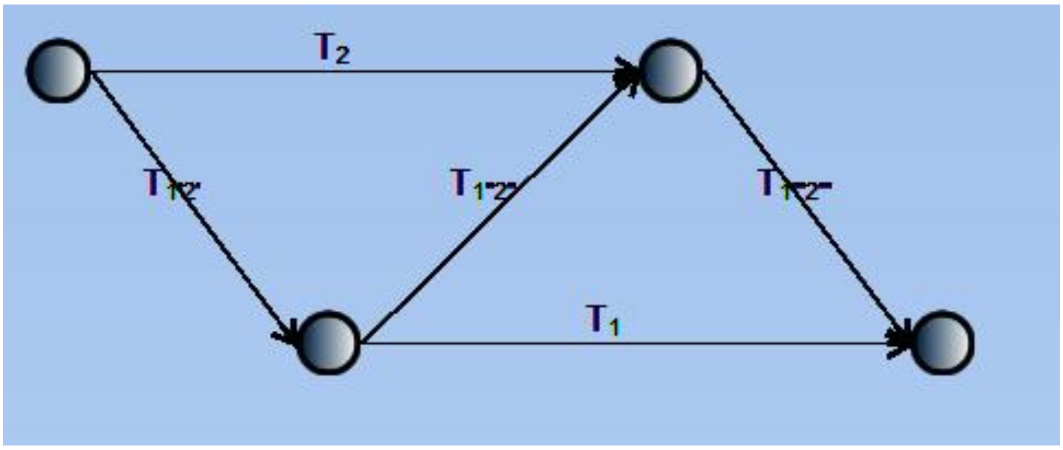


Figure 3.14 Overlapped-by (T_1, T_2)

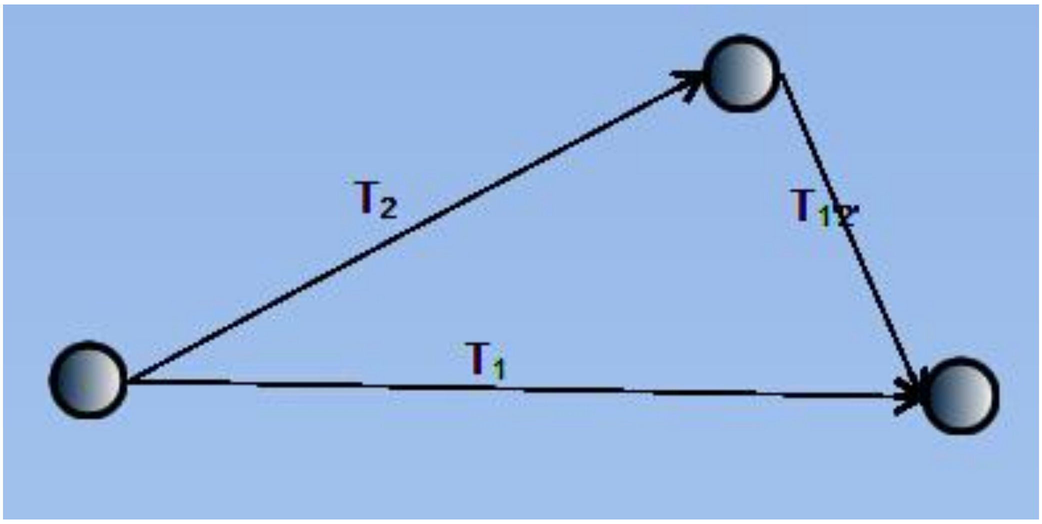


Figure 3.15 Started-by (T_1, T_2)

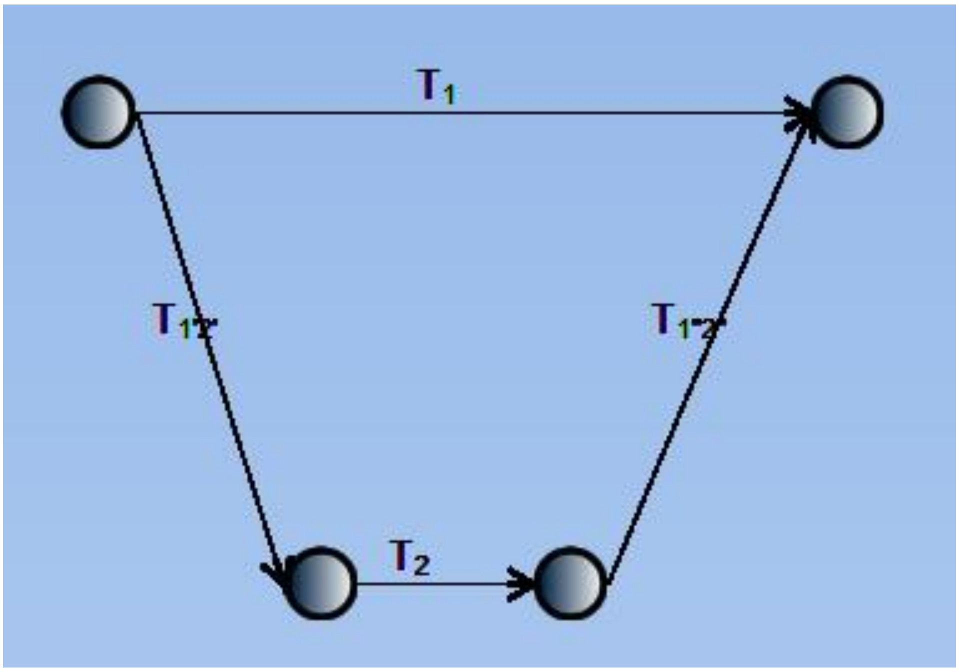


Figure 3.16 Contains (T_1, T_2)

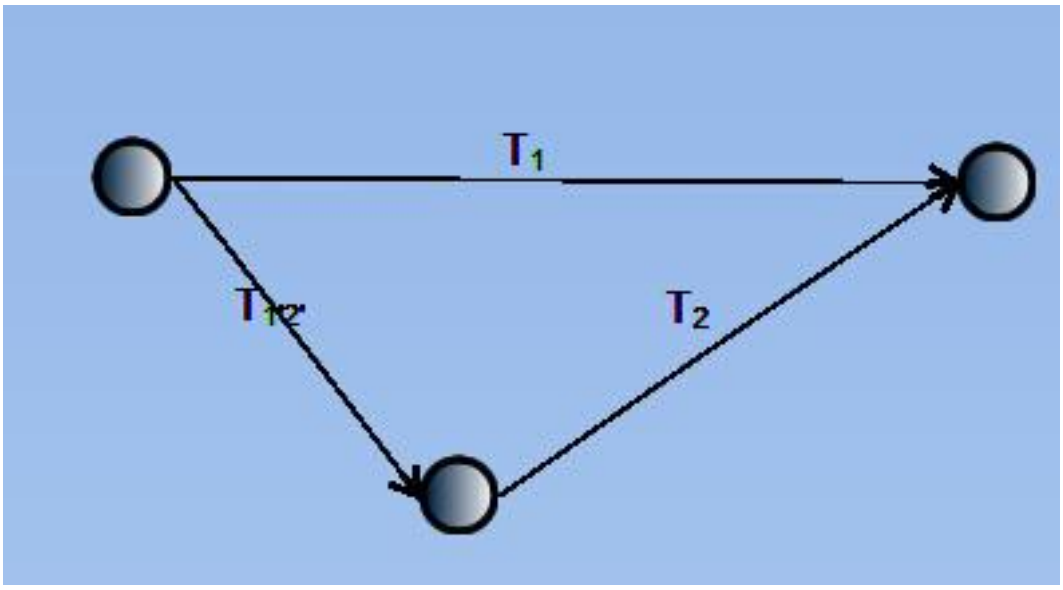


Figure 3.17 Finished-by (T_1, T_2)

Section 3.2 Logical Operation

In standard propositional logic, we can take one of "∧" or "∨" operators together with negation "¬" as primitive operators, which are sufficient to derive all the other logical operators. In addition, as shown in Section 2.1.3, between any two time-elements, there are in total 13 possible relations that are exclusive to each other, that is:

$$\forall t_1,t_2(\text{Equal}(t_1, t_2) \vee \text{Before}(t_1, t_2) \vee \text{After}(t_1, t_2) \vee \text{Meets}(t_1, t_2) \vee \text{Met-by}(t_1, t_2)$$
$$\vee\ \text{Overlaps}(t_1, t_2) \vee \text{Overlapped-by}(t_1, t_2) \vee \text{Starts}(t_1, t_2) \vee \text{Started-by}(t_1, t_2)$$
$$\vee\ \text{During}(t_1, t_2) \vee \text{Contains}(t_1, t_2) \vee \text{Finishes}(t_1, t_2) \vee \text{Finished-by}(t_1, t_2))$$

Therefore, we have the following equivalences:

D1: $\forall t_1, t_2$ (¬ Equal(t_1, t_2) ⇔ Before(t_1, t_2) ∨ After(t_1, t_2) ∨ Meets(t_1, t_2) ∨ Met-by(t_1, t_2) ∨ Overlaps(t_1, t_2) ∨ Overlapped-by(t_1, t_2) ∨ Starts(t_1, t_2) ∨ Started-by(t_1, t_2) ∨during(t_1, t_2) ∨ Contains(t_1, t_2) ∨ Finishes(t_1, t_2) ∨ Finished-by(t_1, t_2)

D2: $\forall t_1, t_2$ (¬ Before (t_1, t_2) ⇔ Equal (t_1, t_2) ∨ After(t_1, t_2) ∨ Meets(t_1, t_2) ∨ Met-by(t_1, t_2) ∨Overlaps(t_1, t_2) ∨ Overlapped-by(t_1, t_2) ∨ Starts(t_1, t_2) ∨ Started-by(t_1, t_2) ∨During(t_1, t_2) ∨ Contains(t_1, t_2) ∨ Finishes(t_1, t_2) ∨ Finished-by(t_1, t_2)

D3: $\forall t_1, t_2$ (¬ After (t_1, t_2) ⇔ Before(t_1, t_2) ∨ Equal (t_1, t_2) ∨ Meets(t_1, t_2) ∨ Met-by(t_1, t_2) ∨Overlaps(t_1, t_2) ∨ Overlapped-by(t_1, t_2) ∨ Starts(t_1, t_2) ∨ Started-by(t_1, t_2) ∨During(t_1, t_2) ∨ Contains(t_1, t_2) ∨ Finishes(t_1, t_2) ∨ Finished-by(t_1, t_2)

D4: $\forall t_1, t_2$ (¬ Meets (t_1, t_2) ⇔ Before(t_1, t_2) ∨ After(t_1, t_2) ∨ Equal (t_1, t_2) ∨ Met-by(t_1, t_2) ∨Overlaps(t_1, t_2) ∨ Overlapped-by(t_1, t_2) ∨ Starts(t_1, t_2) ∨ Started-by(t_1, t_2) ∨During(t_1, t_2) ∨ Contains(t_1, t_2) ∨ Finishes(t_1, t_2) ∨ Finished-by(t_1, t_2)

D5: $\forall t_1, t_2$ (¬ Met-by (t_1, t_2) ⇔ Before(t_1, t_2) ∨ After(t_1, t_2) ∨ Meets(t_1, t_2) ∨ Equal (t_1, t_2) ∨Overlaps(t_1, t_2) ∨ Overlapped-by(t_1, t_2) ∨ Starts(t_1, t_2) ∨ Started-by(t_1, t_2) ∨During(t_1, t_2) ∨ Contains(t_1, t_2) ∨ Finishes(t_1, t_2) ∨ Finished-by(t_1, t_2)

D6: $\forall t_1, t_2$ (¬ Overlaps (t_1, t_2) ⇔ Before(t_1, t_2) ∨ After(t_1, t_2) ∨ Meets(t_1, t_2) ∨ Met-by(t_1, t_2) ∨ Equal (t_1, t_2) ∨ Overlapped-by(t_1, t_2) ∨ Starts(t_1, t_2) ∨ Started-by(t_1, t_2) ∨During(t_1, t_2) ∨ Contains(t_1, t_2) ∨ Finishes(t_1, t_2) ∨ Finished-by(t_1, t_2)

D7: $\forall t_1, t_2$ (¬ Overlapped-by (t_1, t_2) ⇔ Before(t_1, t_2) ∨ After(t_1, t_2) ∨ Meets(t_1, t_2) ∨ Met-by(t_1, t_2) ∨Overlaps(t_1, t_2) ∨ Equal (t_1, t_2) ∨ Starts(t_1, t_2) ∨ Started-by(t_1, t_2) ∨During(t_1, t_2) ∨ Contains(t_1, t_2) ∨ Finishes(t_1, t_2) ∨ Finished-by(t_1, t_2)

D8: $\forall t_1,t_2$ (¬ Starts (t_1, t_2) ⇔ Before(t_1, t_2) ∨ After(t_1, t_2) ∨ Meets(t_1, t_2) ∨ Met-by(t_1, t_2) ∨Overlaps(t_1, t_2) ∨ Overlapped-by(t_1, t_2) ∨ Equal (t_1, t_2) ∨ Started-by(t_1, t_2) ∨During(t_1, t_2) ∨ Contains(t_1, t_2) ∨ Finishes(t_1, t_2) ∨ Finished-by(t_1, t_2)

D9: $\forall t_1,t_2$ (¬ Started-by (t_1, t_2) ⇔ Before(t_1, t_2) ∨ After(t_1, t_2) ∨ Meets(t_1, t_2) ∨ Met-by(t_1, t_2) ∨Overlaps(t_1, t_2) ∨ Overlapped-by(t_1, t_2) ∨ Starts(t_1, t_2) ∨ Equal (t_1, t_2) ∨During(t_1, t_2) ∨ Contains(t_1, t_2) ∨ Finishes(t_1, t_2) ∨ Finished-by(t_1, t_2)

D10: $\forall t_1,t_2$ (¬ During (t_1, t_2) ⇔ Before(t_1, t_2) ∨ After(t_1, t_2) ∨ Meets(t_1, t_2) ∨ Met-by(t_1, t_2) ∨Overlaps(t_1, t_2) ∨ Overlapped-by(t_1, t_2) ∨ Starts(t_1, t_2) ∨ Started-by(t_1, t_2) ∨ Equal (t_1, t_2) ∨ Contains(t_1, t_2) ∨ Finishes(t_1, t_2) ∨ Finished-by(t_1, t_2)

D11: $\forall t_1,t_2$ (¬ Contains (t_1, t_2) ⇔ Before(t_1, t_2) ∨ After(t_1, t_2) ∨ Meets(t_1, t_2) ∨ Met-by(t_1, t_2) ∨Overlaps(t_1, t_2) ∨ Overlapped-by(t_1, t_2) ∨ Starts(t_1, t_2) ∨ Started-by(t_1, t_2) ∨During(t_1, t_2) ∨ Equal (t_1, t_2) ∨ Finishes(t_1, t_2) ∨ Finished-by(t_1, t_2)

D12: $\forall t_1,t_2$ (¬ Finishes (t_1, t_2) ⇔ Before(t_1, t_2) ∨ After(t_1, t_2) ∨ Meets(t_1, t_2) ∨ Met-by(t_1, t_2) ∨Overlaps(t_1, t_2) ∨ Overlapped-by(t_1, t_2) ∨ Starts(t_1, t_2) ∨ Started-by(t_1, t_2) ∨During(t_1, t_2) ∨ Contains(t_1, t_2) ∨ Equal (t_1, t_2) ∨ Finished-by(t_1, t_2)

D13: $\forall t_1,t_2$ (¬ Finished-by (t_1, t_2) ⇔ Before(t_1, t_2) ∨ After(t_1, t_2) ∨ Meets(t_1, t_2) ∨ Met-by(t_1, t_2) ∨Overlaps(t_1, t_2) ∨ Overlapped-by(t_1, t_2) ∨ Starts(t_1, t_2) ∨ Started-by(t_1, t_2) ∨During(t_1, t_2) ∨ Contains(t_1, t_2) ∨ Finishes(t_1, t_2) ∨ Equal(t_1, t_2)

Thus, in the case of this research work, we can take "∧" and "∨"as primitive connectives, which are sufficient to express all the logical combination of temporal relations.

Conclusion:

C1) all the temporal relations can be expressed by the conjunction of the single Meets relation;

C2) all the logical connectives among Meets table can be expressed by logical connective "$\wedge$" and "$\vee$".

One problem comes that how to graphically represent logical connective "$\vee$". Section 3.3 will introduce the graphical representation for uncertain or incomplete temporal knowledge.

Section 3.3 Graphically Representing Uncertain And Incomplete Temporal Knowledge

On the one hand, for a given pair of time elements t1 and t2, it may be unknown which of the 30 possible temporal relations, as classified in Section 2.1.2, certainly holds between t1 and t2. We shall formalize this uncertain temporal knowledge in term of temporal relations joint by disjunctive connectives. On the other hand, for a given situation, the corresponding temporal knowledge of what time elements are involved, and what are the exact durations of these time elements, may be only partially known.

In general, the temporal order relation between two time elements may be given in any form of those 30, as introduced in Section 2.1.3. However, each of these temporal relations can be defined in terms of the single Meets relation. Therefore, all the knowledge about the temporal relations over a given collection of time elements (points and/or intervals) can be transformed and stored as a table of Meets relations in the knowledge base.

Following the approach of [JBMX2010], we shall use a triad, (T, M, D), to express the temporal reference of a given collection of temporal knowledge, where:

- T = $\{t_1, \ldots, t_n\}$ is a finite set of time elements, expressing the knowledge (possibly incomplete) of what time elements are involved;
- M = $\{Meets(t_i, t_{i(1)}) \vee \ldots \vee Meets(t_i, t_{i(j)}) \mid$ for some i, where $1 \leq i, i(1), i(j), j \leq n\}$ is a collection (i.e., conjunction) of disjunctions of Meets relations over T, expressing the knowledge (possibly incomplete) as to how the time elements in T are related to each other by the Meets relations;
- D = $\{Dur(t_i) = r_i \mid$ for some i where $1 \leq i \leq n\}$ is a collection of duration assignments (possibly incomplete) to time elements in T.

In addition to the semantic interpretations (S1. – S4.), provided at the beginning of this chapter, as an extension to the work of [JBMX2010], we propose the following additional ones:

S5. Meets(T_i, T_j) $\vee$ Meets(T_i, T_k) is denoted by defining T_i as duplicated identical out-arcs of a node, and defining one of the two T_is as an in-arc and T_j as an out-arc of another node; and defining the other T_i as an in-arc and T_k as an out-arc of the third node (See Figure 3.18).

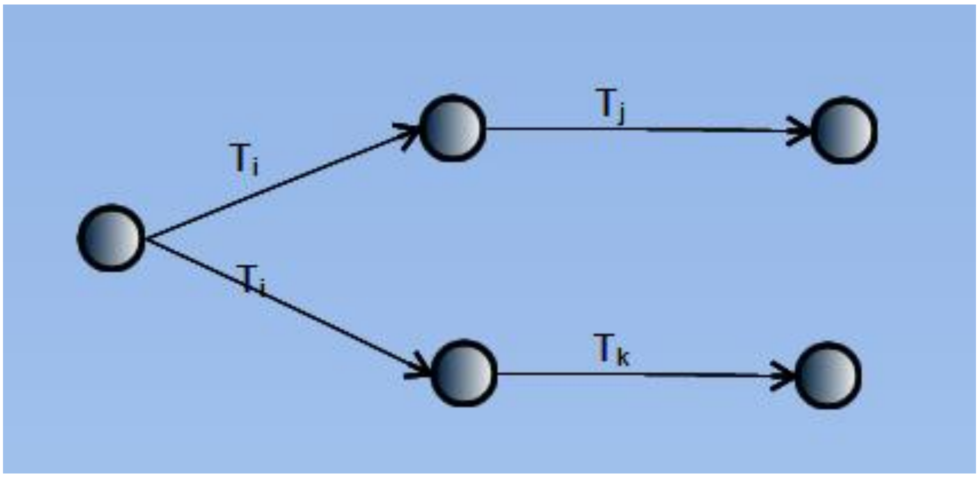


Figure 3.18 Meets(T_i, T_j) $\vee$ Meets(T_i, T_k)

S6. Meets(T_i, T_k) $\vee$ Meets(T_j, T_k) is denoted by defining T_k as duplicated identical in-arcs of a node, and defining T_i as an in-arc and one T_k as an out-arc of another node; and defining T_j as as an in-arc and the other T_k as an out-arc of the third node respectively (See Figure 3.19).

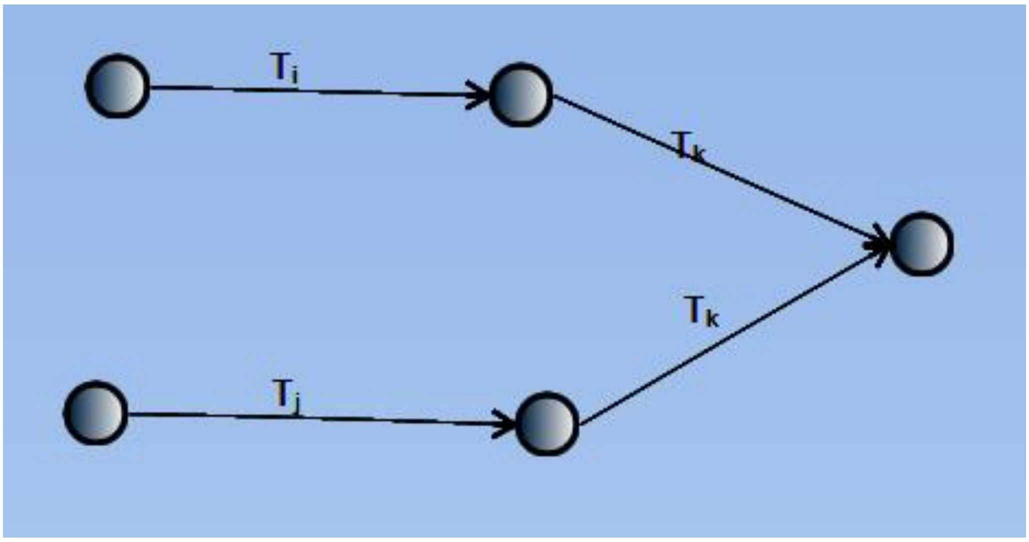


Figure 3.19 Meets(T_i, T_k) $\vee$ Meets(T_j, T_k)

What follows are expressions of the combinations of “AND” and “OR”:

E1: First “AND” then “OR” (See Figure 3.20 and Figure 3.21)

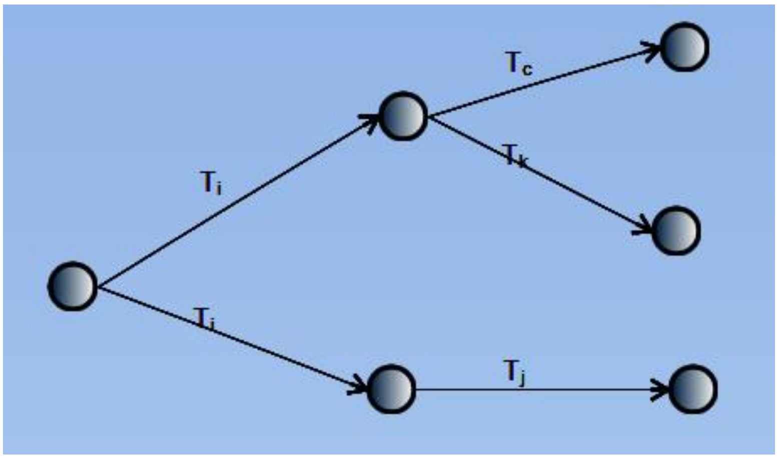


Figure 3.20 Meets(T_i,T_k)∧Meets(T_i,T_c)∨Meets(T_i,T_j)

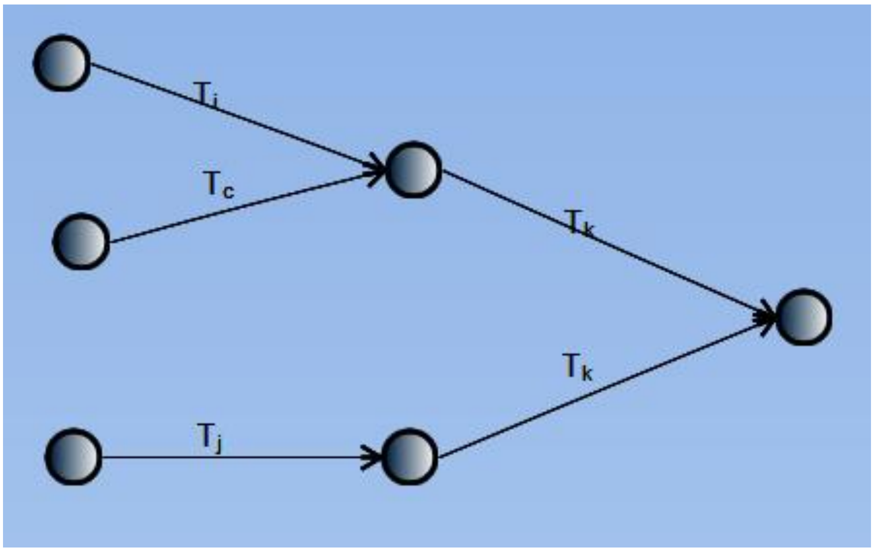


Figure 3.21 Meets(T_i,T_k)∧Meets(T_c,T_k)∨Meets(T_j,T_k)

E2: First "OR" then "AND" – Distribution (See Figure 3.22 and Figure 3.23)

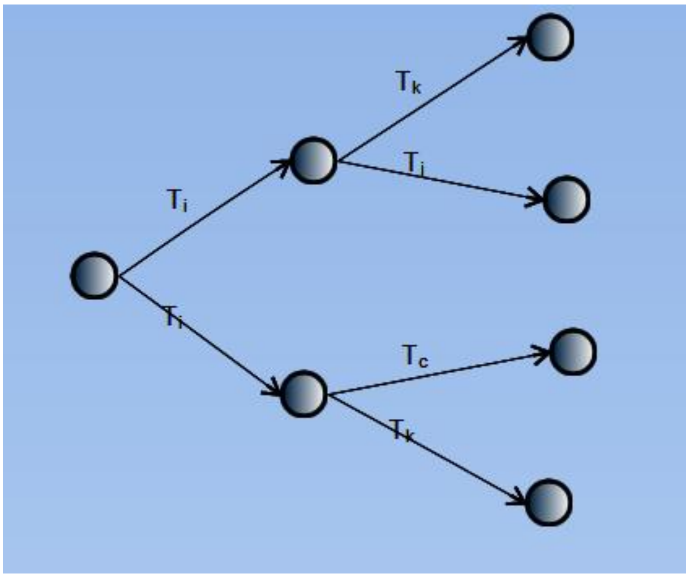


Figure 3.22 Meets(T_i,T_k)∧(Meets(T_i,T_c) ∨Meets(T_i,T_j))

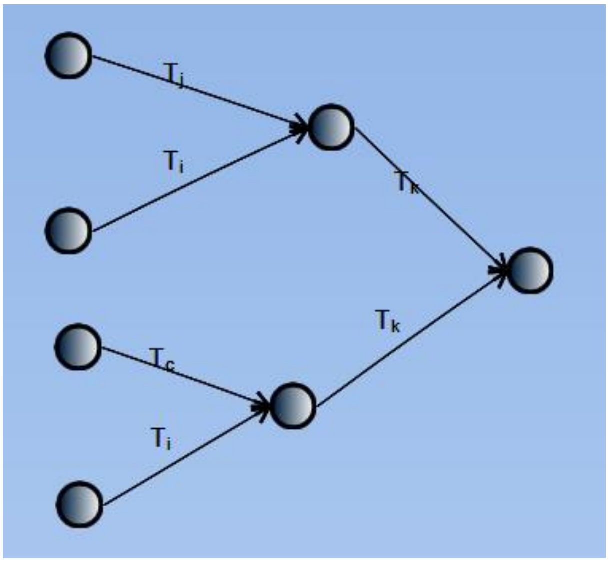


Figure 3.23 Meets(Ti,Tk)∧(Meets(Tc,Tk)∨Meets(Tj,Tk))

E3: Conjunction of Disjunctions (See Figure 3.24)

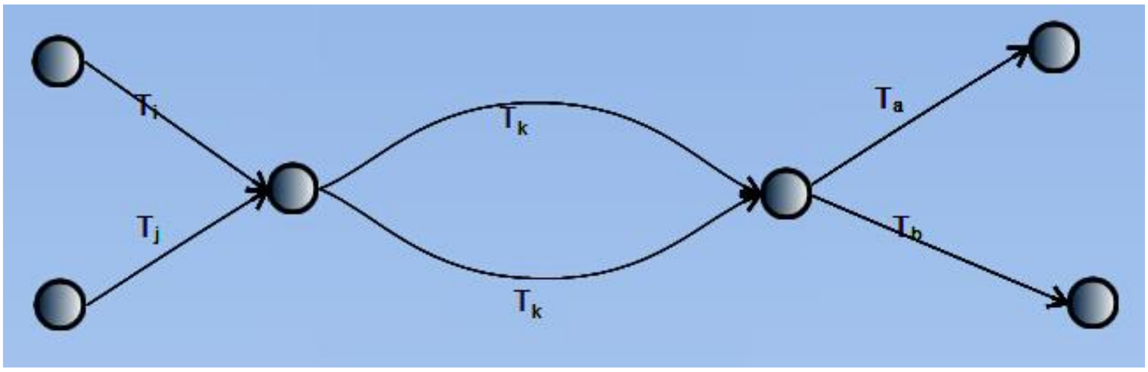


Figure 3.24 $(Meets(T_i,T_k) \vee Meets(T_j,T_k)) \wedge (Meets(T_k,T_a) \vee Meets(T_k,T_b))$

Section 3.4 An Example Of Time Graph

As an example, a (T1, M1, D1) is shown as follows:

$T1 = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}\}$

$M1 = \{Meets(t_1,t_2), (Meets(t_2,t_3) \vee Meets(t_2,t_4)),$

$Meets(t_3,t_7)$, $Meets(t_3,t_8)$, $Meets(t_4,t_5)$, $Meets(t_4,t_6)$, $Meets(t_5,t_7)$, $Meets(t_5,t_8)$, $Meets(t_7,t_9)$, $Meets(t_7,t_{10})$, $Meets(t_8,t_{12})$,$Meets(t_6,t_{11}) \vee Meets(t_9,t_{11}) \vee Meets(t_{10},t_{11}))$, $Meets(t_{11},t_{12})\}$

$D1 = \{Dur(t_2)=1, Dur(t_3)=2.3, Dur(t_6)=3, Dur(t_8)=2.6, Dur(t_{10})=0.8, Dur(t_{12})=1\}$

where Figure 3.25 shows the corresponding temporal graph .

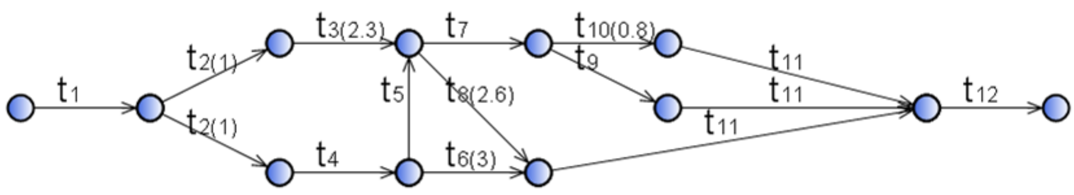


Figure 3.25 A Temporal Graph Example

A temporal scenario is defined as a maximal sub-graph of given temporal graph with no duplicated time elements. The six temporal scenarios of (T1, M1, D1) are shown as follows: (See Figure 3.26 - Figure 3.31).

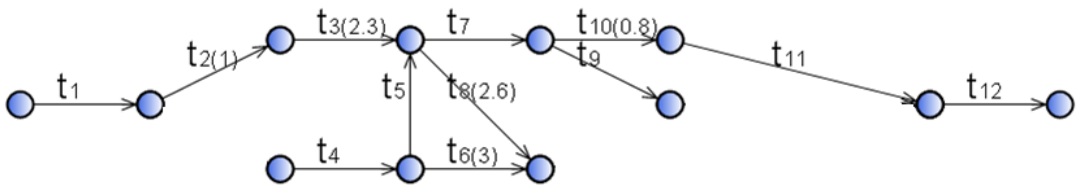


Figure 3.26 Temporal Scenarios 1

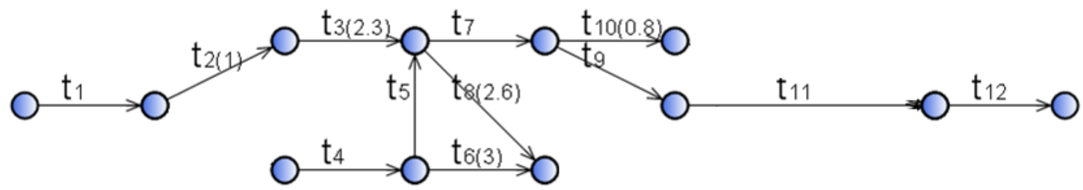


Figure 3.27 Temporal Scenarios 2

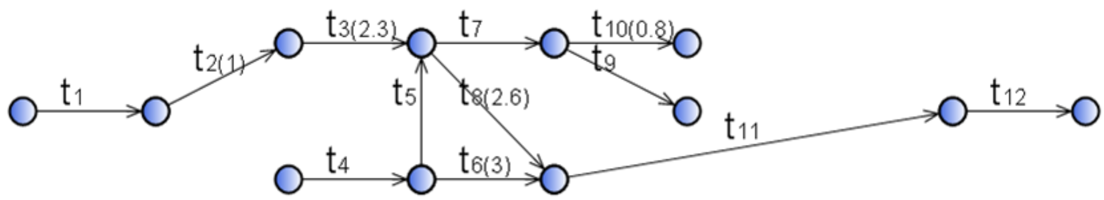


Figure 3.28 Temporal Scenarios 3

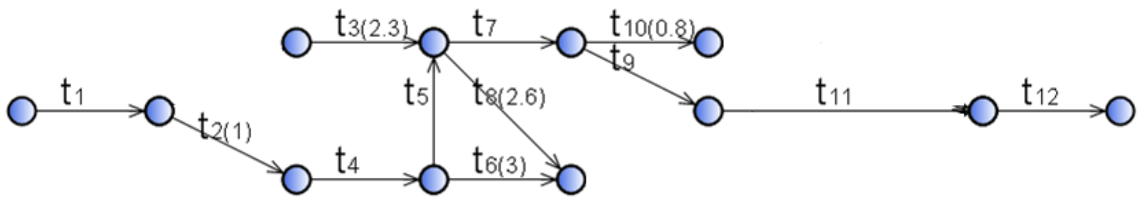


Figure 3.29 Temporal Scenarios 4

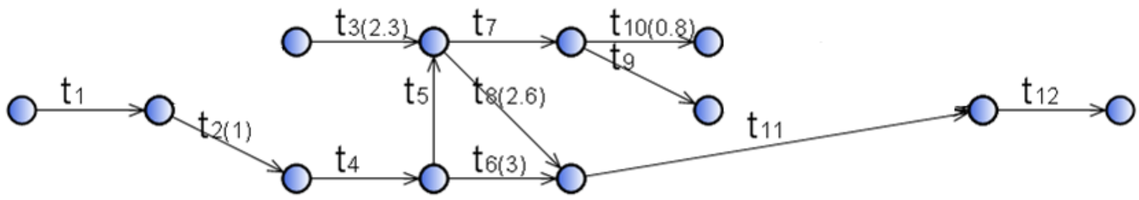


Figure 3.30 Temporal Scenarios 5

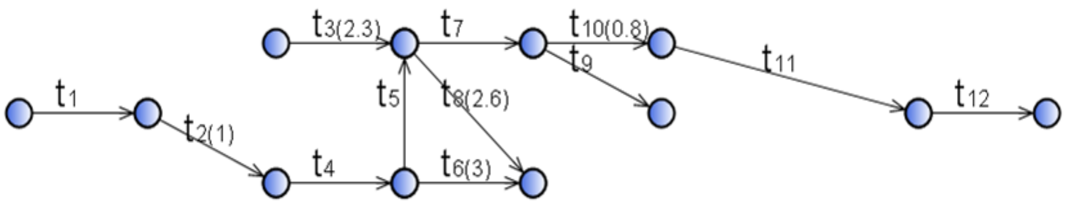


Figure 3.31 Temporal Scenarios 6

Section 3.5 Consistency Of Temporal Knowledge

A temporal reference (T, M, D) is defined as temporal consistent if at least one of its temporal scenarios is temporal consistent. The necessary and sufficient conditions for the consistency of a scenario can be given as below:

1) For each simple circuit in a scenario, the directed sum of weights is zero;

2) For any two adjacent time elements, the directed sum of their weights is bigger than zero.

Here, condition 1) guarantees that there exists a valid duration assignment function Duration to the time elements in scenario consistent with D (See Figure 3.32); and condition 2) ensures that no two time points can meet each other, that is between any two time points, there exists a time interval between them (See Figure 3.33).

1) $Dur(T_2) < Dur(T_1) + Dur(T_{1'2'}) + Dur(T_{1''2''})$ is inconsistent (See Figure 3.32).

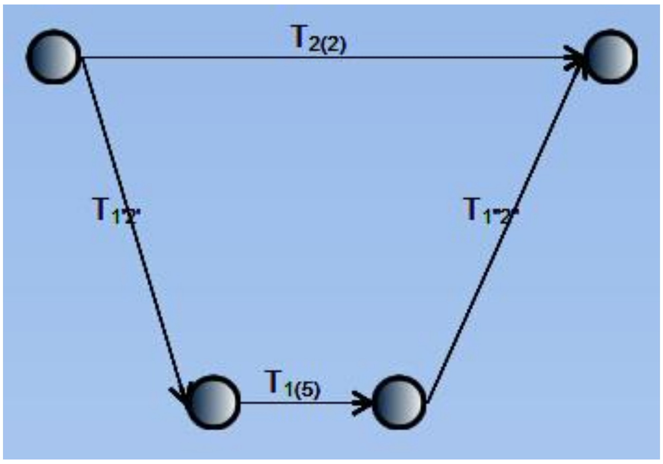


Figure 3.32 $Dur(T_2) < Dur(T_1) + Dur(T_{1'2'}) + Dur(T_{1''2''})$

2) $Dur(T_1) + Dur(T_2) = 0$ is inconsistent.(See Figure 3.33)

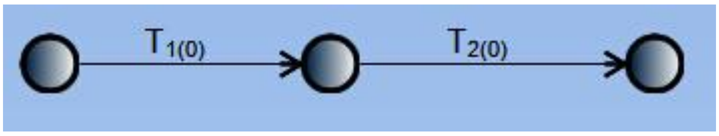


Figure 3.33 $Dur(T_1) + Dur(T_2) = 0$

The consistency checking for a temporal scenario with duration constraints involves searching for simple circuits, and constructing a numerical constraint for each circuit. The existence of a solution(s) to this set of constraints implies the consistency of the temporal scenario and hence of the temporal reference, where each solution gives a possible case that can subsume the addressed temporal scenario. However, this thesis will transform the consistency checking problem for temporal references into linear programming problem.

For example, consider a given scenario:

$$Meets(T_1,T_4) \wedge Meets(T_1,T_2) \wedge Meets(T_1,T_6) \wedge$$

$$Meets(T_4,T_5) \wedge Before(T_2,T_5) \wedge Finishes(T_5,T_6)$$

$$Dur(T_1) = 1, Dur(T_2) = 3, Dur(T_6) = 6$$

It can be expressed as follows:

$T = \{T_1, T_2, T_4, T_5, T_6, T_{2'5'}\}$;

$M = \{Meets(T_1,T_4), Meets(T_1,T_2), Meets(T_1,T_6), Meets(T_4,T_5), Meets(T_2,T_{2'5'}), Meets(T_{2'5'},T_5)\}$;

$D = \{ Dur(T_1) = 1, Dur(T_2) = 3, Dur(T_6) = 6 \}$;

where Figure 3.34 shows the corresponding graph

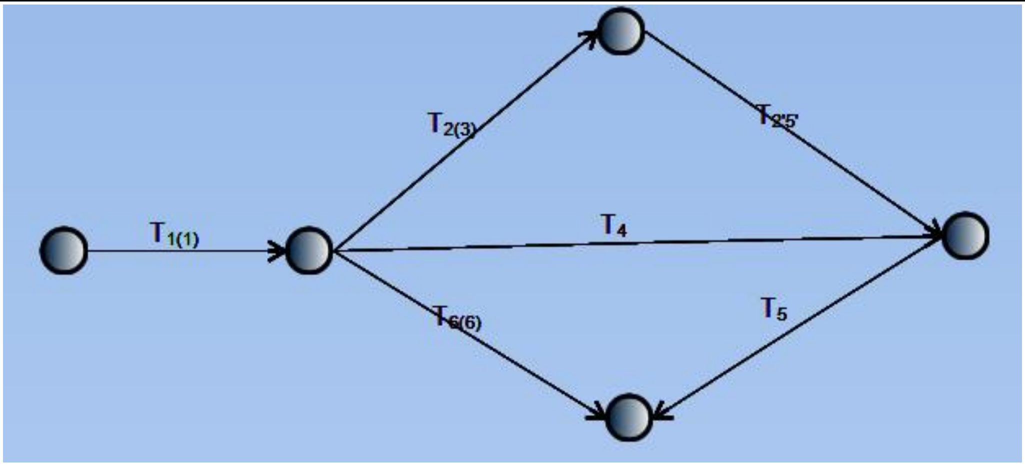


Figure 3.34 A Sample (T, M, D)

then:

$$Dur(T_4) = Dur(T_{2'5'}) + 3\ ;$$

$$6 = Dur(T_4) + Dur(T_5);$$

$$3 + Dur(T_{2'5'}) + Dur(T_5) = 6;$$

$$Dur(T_4) \geq 0\ ,\ Dur(T_5) \geq 0,\ Dur(T_{2'5'}) \geq 0;$$

then define $Dur(T_4) = x$, $Dur(T_5) = y$, $Dur(T_{2'5'}) = z$, then

$$\begin{cases} x & + & y & = & 6 \\ x & - & z & = & 3 \\ 3 & + & y & +z & =6 \end{cases}$$

$$\Rightarrow \left[\begin{array}{ccc:c} 1 & 1 & 0 & 6 \\ 1 & 0 & -1 & 3 \\ 0 & 1 & 1 & 3 \end{array}\right]$$

where we can use a linear programming algorithm to check the consistency of PIBS:

Consistent(0≦x≦3, 0 ≦ y ≦ 3)

which means that it will be consistent when

(0 ⩽ x ⩽3) ∧ (0⩽ y ⩽3).

CHAPTER 4 VISUAL TIME

In this chapter, a computer aided case tool, called Visual Time, is introduced. It provides a user-friendly environment for online social media searching, filtering and describing scenarios with rich temporal information in natural language, which can be modelled in terms of Time Relationship Diagrams (TRD). The correspondence between sentences (and/or key phrases) in a given temporal scenario and the corresponding elements of the time relationship diagram can be automatically identified in a visualized way. From a time relationship diagram, the corresponding relative temporal relations and the equivalent "Meets" table can be derived automatically. A TRD can be automatically and visually transformed into its corresponding Time Graph, which is supported by an automatic consistency checker that delivers a visual and audio verdict as to whether a given scenario is temporally consistent or not.

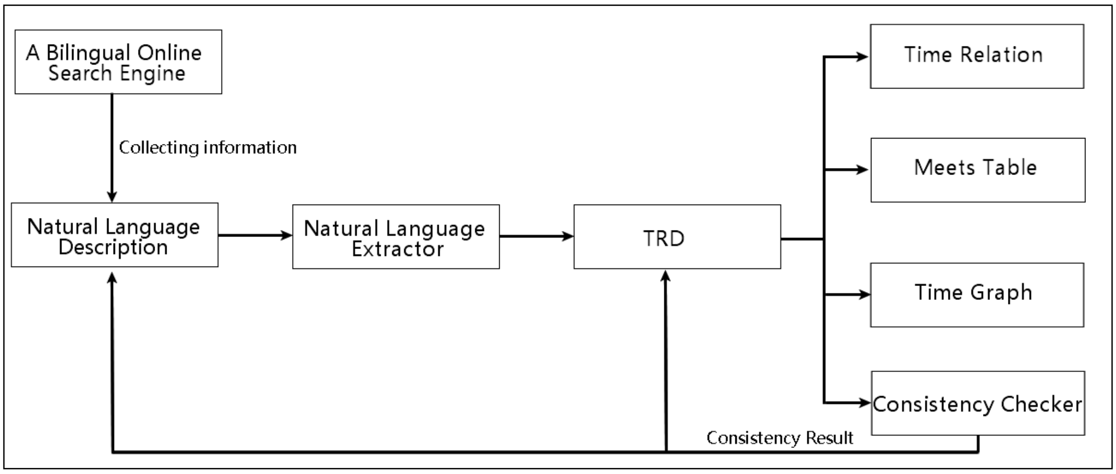


Figure 4.1 Structure of Visual Time

Figure 4.1shows the structure of this thesis's framework.

Section 4.1 A Social Media Online Search Engine

Social media searching is quite different with normal text searching because of the following reasons:

1. Social media users may have different roles. For example, if one is at his office, he is a worker; but if he/she is at home, he/she maybe a father/mother. If he/she often appears at certain airport station, he/she maybe work for Airline Company; but if he/she often appears at different airport station, he/she maybe a flight attendant or a man who often has business travelling. However, this is very important for both the advertising company and temporal logic research. Especially, many social media users choose to close the GPS locating model of mobile phone because of personal security reasons, they cannot close the time model. This means that every tweet must have its time label.

2. Social media users will not use long contexts. This means the traditional search algorithms like PageRank [XG2004], HITS (Hyperlink-Induced Topic Search) [CPH2008] will not be adequate for it. Moreover, all of the algorithms that search by TF (Term Frequency) [RAJ2011] are also not suitable here. This is because TF determines the importance level according to the frequencies of keywords while social media users only care about the number of their friends and fans.

3. Social media are unfiltered. Over 200 million tweets are sent every day, over 1 billion Facebook users login their accounts every week. A huge amount of data runs online, but most of it is unfiltered information.

Figure 4.2 shows the main user interface of social media searching engine. Users can input the city name by clicking button "Input location" to quickly locate the city they want to search. Then users need to draw a circle over the specified area.

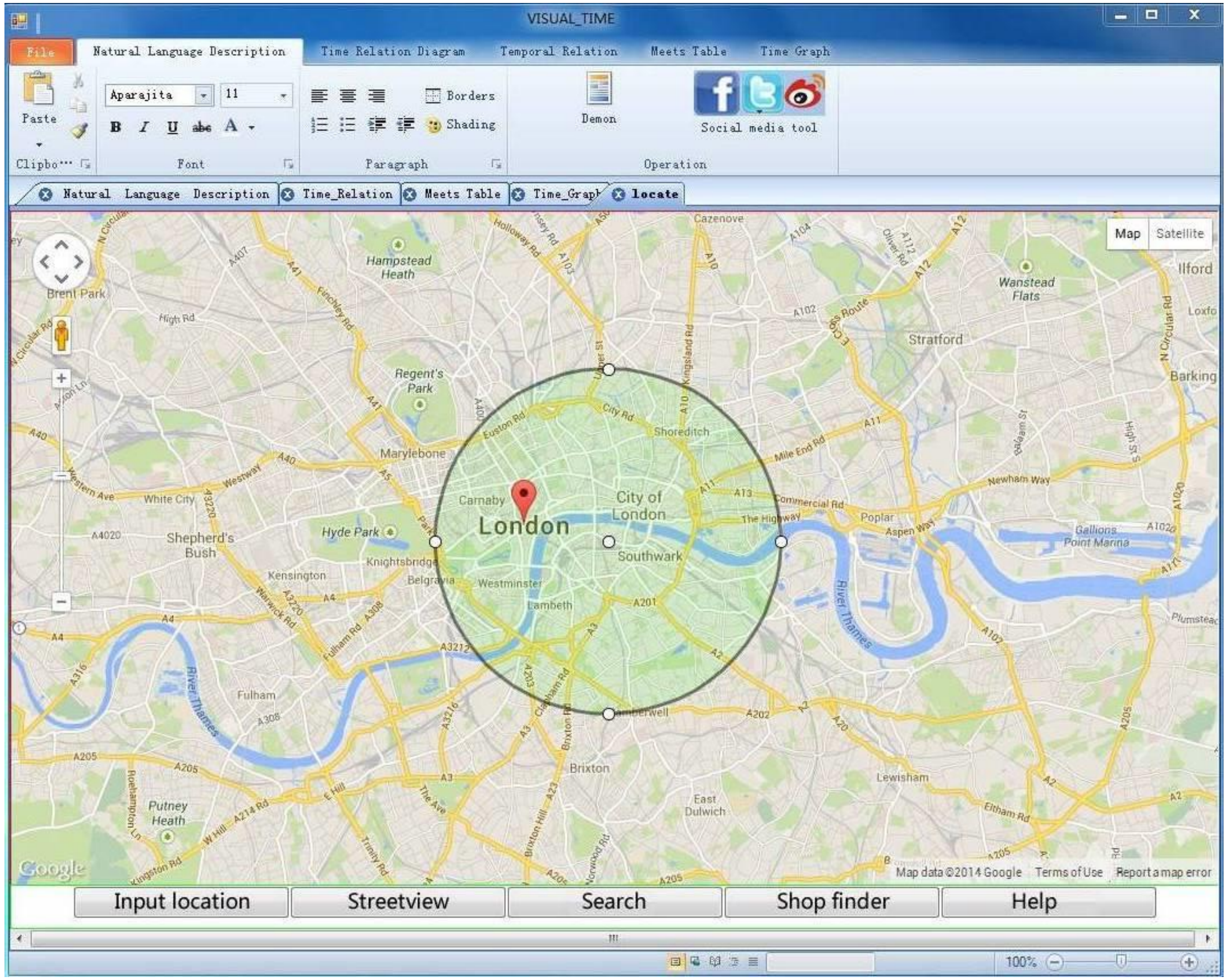


Figure 4.2 Main User Interface

Figure 4.3 Search Result

After clicking the button "**Search**", all the social media users within the chosen circle will be automatically shown in a new webpage (See Figure 4.3). Users can choose some social media users by clicking the **profile picture** of users, chosen social media users will be automatically recorded into **chosen users list**. Then users can send private message to all the members of **Chosen users for filtering** list**.** Figure 4.4 and Figure 4.5 are the corresponding Chinese version of user interface.

Figure 4.4 Chinese User Interface

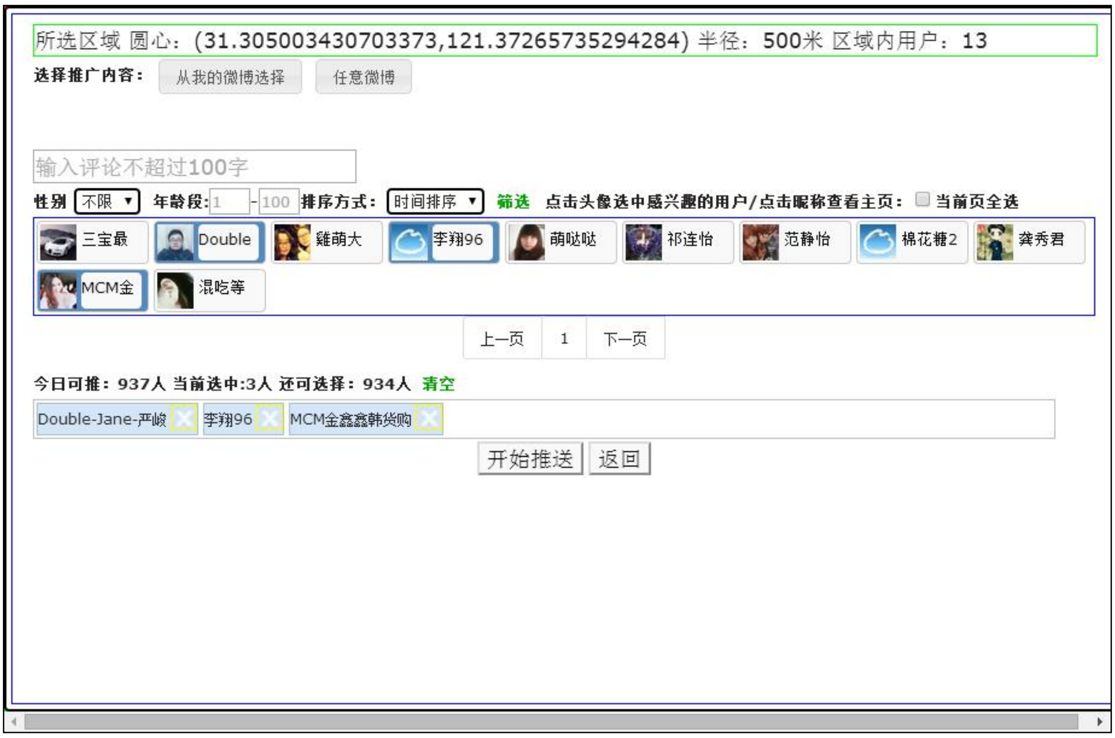


Figure 4.5 Chinese Search Result

Section 4.1.1 Social media filtering algorithm

This social media filtering algorithm is designed to filter social media users and classifies these users into three levels - **hot users, potential users, frozen users**. The corresponding pseudocode is shown as follows:

N.B.

1] **The data structure of Social media user list** is a heap, the social media search engine is keeping on collect social media users from social media platforms;

2] **X** is a time section, which is set by user. For example, if one wants to research the activity of one social media user within sixty days, he/she can set X as sixty.

3] **σ-standard** is a threshold which is set by user. **σ-standard** is designed to classify the social media users. Take KFC for example, **hot users** are the ones who often send tweets(s) (is larger than σ-standard) in KFC within X days; **potential users** are the ones who send little tweet(s) (σ is smaller than σ-standard) in KFC within X days; **frozen users** are the ones who have never sent tweets in KFC.

Input: Social media user list;

Output: Frozen user list (initialized as empty), **Potential user** list (initialized as empty), **Hot user** list (initialized as empty);

Begin

While (the length of Social media user list < 1)

{

Collect the first user A from social media user list;

Else

User set X;

Save the time of latest Tweet sent by A as L;

Save the time of now as T1;

If T1 – L > X

Add A in **Frozen user** list;

Delete A from social media user list;

If **Potential user** list contains A, delete A from **Potential user** list;

If **Hot user** list contains A, delete A from **Hot user** list;

Else

Save the register time of A as (T0);

Save the time of every tweet sent by A within X days as Di (i ∈ 0...n);

Save the number of tweets sent by A within X days as N0;

calculate σ;

Set σ-standard;

If σ > σ-standard, save A in **Hot user** list;

If σ≦σ-standard, save A in **Potential user** list;

}

End;

where step **calculate** σ is calculated by the following equations:

$$TD_i = \begin{cases} D_i(i \le n) \\ D_{2n-i}(i > n) \end{cases} (i \in 1 \ldots 2n);$$

$$\text{if } T1 - T0 \ge X \text{ then } x_i = T1 - X + N0/X \times i; (i \in 1 \ldots 2n)$$
$$\text{if } T1 - T0 < X \text{ then } x_i = T1 - T0 + N0/T0 \times i; (i \in 1 \ldots 2n)$$

$$\sigma = \sqrt{\frac{1}{2n-1}\sum_{i=1}^{2n}(TD_i - x_i)^2};$$

This aim of algorithm is to classify social media users into three different classes: hot users, potential users and frozen users. This social media search engine can not only be used for business purpose, but the public security can also be helped. Take a murder case for example: using the algorithm above, the police officer can contact the potential witness around the geographical location of murdering by the order: **hot users -> potential users->frozen users**. Google street view is also supported, which can help users to get the geographical information for deciding which area should be chosen (See Figure 4.6).

Figure 4.6 Street View

Section 4.2 Semi-automatic Temporal Relation Extraction

As what has been discussed in Section 4.1, all the actions and times in natural text need to be extracted. This section addresses the question of how to deal with natural language knowledge. The corresponding pseudocode is shown as follows:

Section 4.2.1 Semi-automatic Temporal Relation Extraction Algorithm

The semi-automatic temporal relation extraction algorithm is designed to extract time element and temporal relations from natural texts. The corresponding pseudocode is shown as follows:

Input: Natural sentence list;
Output: TRD (initialized as empty);

```
Begin
    For int i = 0 to the length of natural sentence list
    {
        Use SUTime to tag natural sentence[i];
        Store all verbs in Verb list;
        Store time in time list;
        Use a given dictionary to extract temporal relation;
        Store temporal relation in temporal relation list;
        Use Verb list and time list to draw time element in TRD;
        Draw the corresponding temporal relation(s) between time elements in TRD;
    }
End;
```

An example about Step **Use a given dictionary to extract temporal relation** is introduced as follows:

Tom got home before 1:00 PM.

where we can illustrate temporal relations "Before (got home, 1:00 PM)".

The above algorithm can extract most of the time elements, except very complicated and long sentences. Because of the limitation of NLP (Natural Language Processing), it has been impossible to obtain the all the 13 temporal relations between each time element. Therefore, the above algorithm is defined as “semi-automatic”, in the sense that although the algorithm can automatically extract most of the actions and times, some other actions and times may need to be extracted manually. This issue will be discussed further in Section 7.1.

Section 4.2.2 Performance Evaluation

This section values the interface of Visual Time. Two examples are used as follows:

Example 1:

I got home before 1:00 PM. And he left after 2:00 PM.

Figure 4.7 shows the corresponding Time Relation Diagram (TRD) of above sentence.

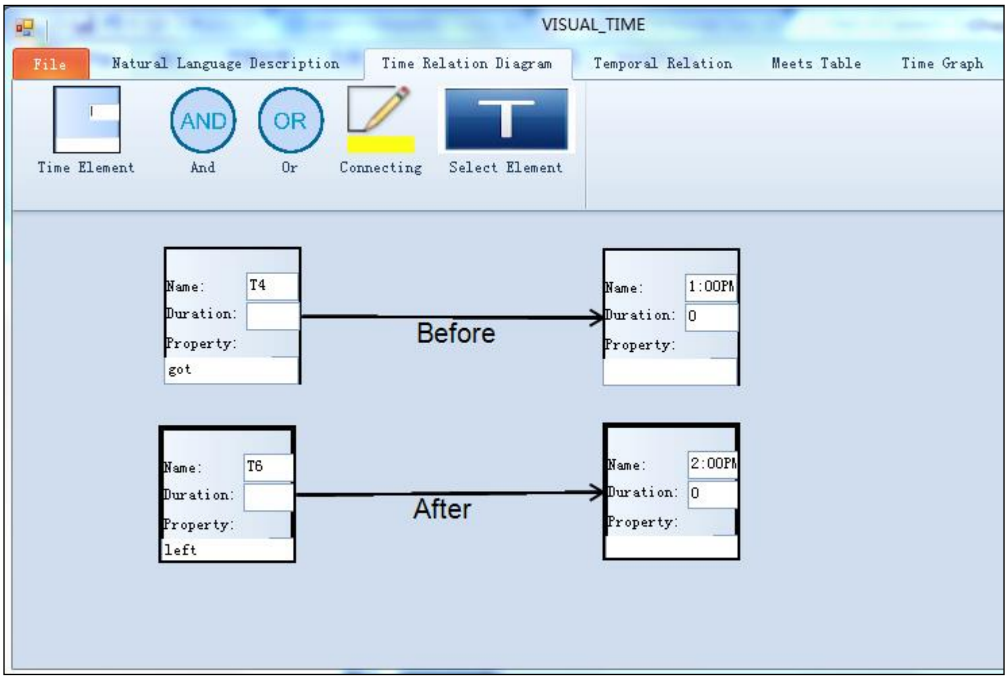


Figure 4.7 Performance 1

Example 2:

I left home at 2:00 PM and went to Peter's house. He was playing a video, and we waited till it finished. Then we went together to the train station and waited for Jack. Jack got to the train station at 4:00 PM.

Figure 4.8 shows the corresponding TRD of above sentence.

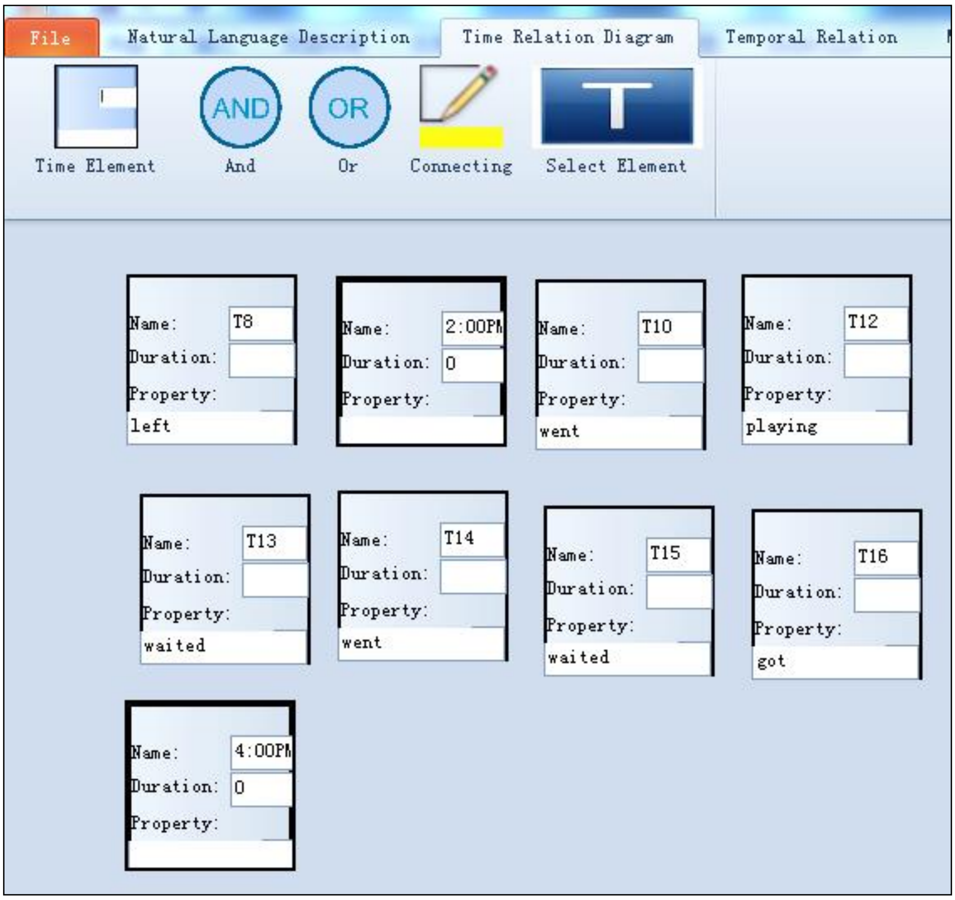


Figure 4.8 Performance 2

SUTime was evaluated on TempEval-2 Task in [Ver2010], which consists of two parts: identifying the extents of a temporal expression and then providing the correct TIMEX3 type and value attributes for each recognized expression. Based on SUTime, this improved algorithm can extract most of time elements.

Section 4.3 Time Relation Diagram (TRD)

[MK1994] introduced a new concept called Time Relation Diagram(TRD), similar to Peter Chen's Entity relationship Diagram (ERD), Time Relation Diagram (TRD) is proposed here, especially for modelling temporal information of given scenarios described in natural language, in which, time elements, duration and temporal relations between time elements can be graphically represented as diagrams.

Section 4.3.1 TRD Model

The representation of time and relationship type has been introduced in chapter 2; however, a TRD is simply a tool to visualize the time relations of uncertain or incomplete temporal information.

Each entity of an entity type is associated with a time element capturing the valid time of the entity. The time element can be a time interval or a time point. A time element of TRD is represented by four main attributes:

1. **Name**: an independent temporal Entities(time elements) that can be uniquely identified, e.g. T1,T2, 1:00 PM;

2. **Duration**: the time length of time elements, e.g. 2 hours, 2 minutes;

3. **Property**: the corresponding actions or times of time elements, e.g. "got", "works";

4. **Sentence**: every time element extracted from natural information must match one or more corresponding natural sentences;

5. **Temporal Relationships** (hidden): Allen's 13 temporal relations which were discussed in Section 2.1.2. The temporal relation is a hidden attribute which will be marked over arrow-lines; (See Figure 4.9)

Figure 4.9 Time Point

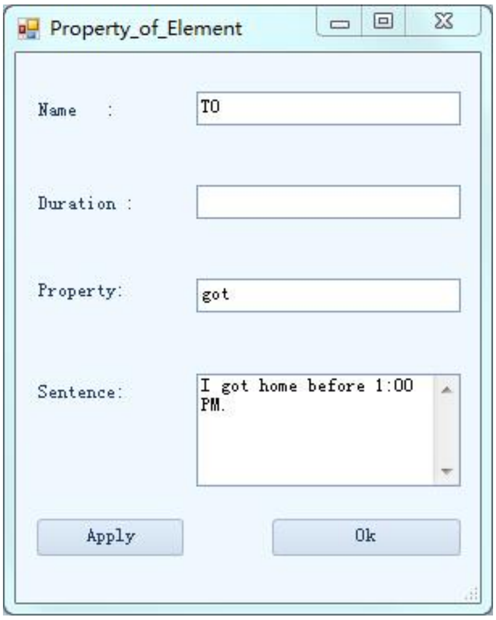


Figure 4.10 Time Interval

Figure 4.9 and Figure 4.10 show examples of time elements. The corresponding TRD of the 7 kinds of basic graph structures expressed in Section 2.2.2 are shown from Figure 4.11 to Figure 4.17.

- Meets(T_i, T_j) $\vee$ Meets(T_i, T_k)   (See Figure 4.11)

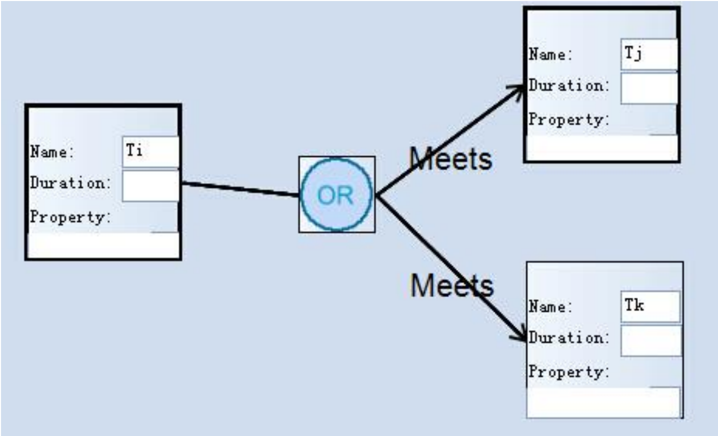


Figure 4.11 Meets(T_i, T_j) $\vee$ Meets(T_i, T_k)

■ Meets(T_i, T_k) ∨ Meets(T_j, T_k) (See Figure 4.12)

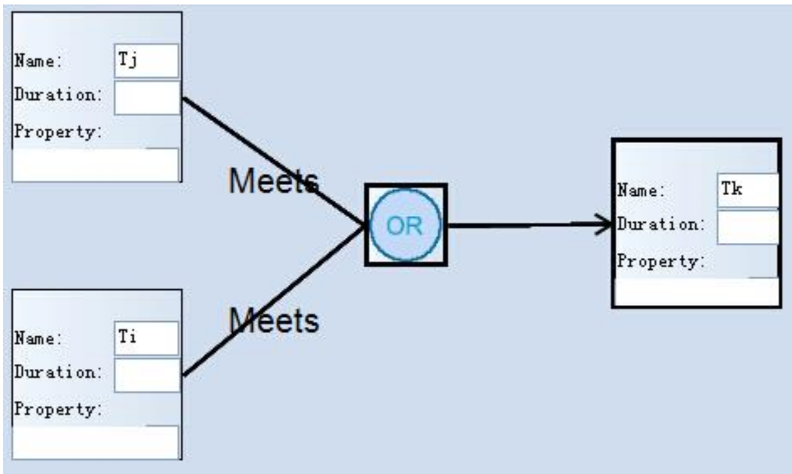


Figure 4.12 Meets(T_i, T_k) ∨ Meets(T_j, T_k)

■ Meets(T_i,T_k)∧Meets(T_c,T_k)∨Meets(T_j,T_k)(See Figure 4.13)

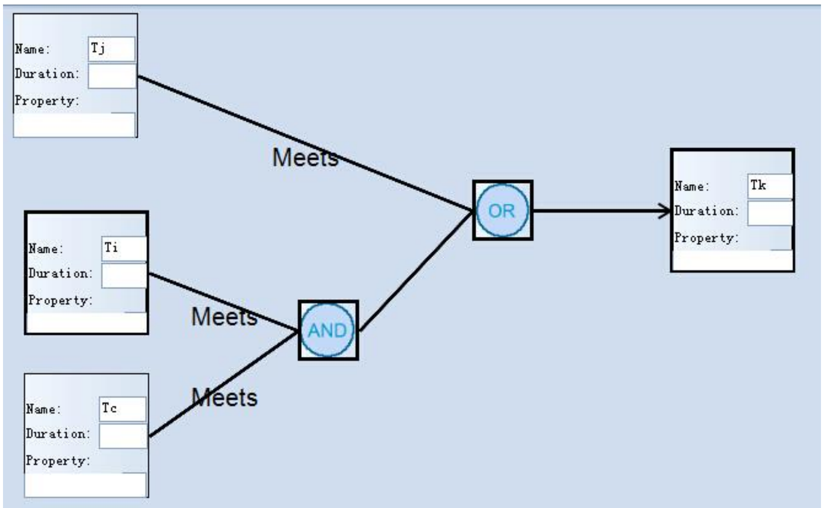


Figure 4.13 Meets(T_i,T_k)∧Meets(T_c,T_k)∨Meets(T_j,T_k)

■ Meets(T_i,T_k)∧Meets(T_i,T_c)∨Meets(T_i,T_j)(See Figure 4.14)

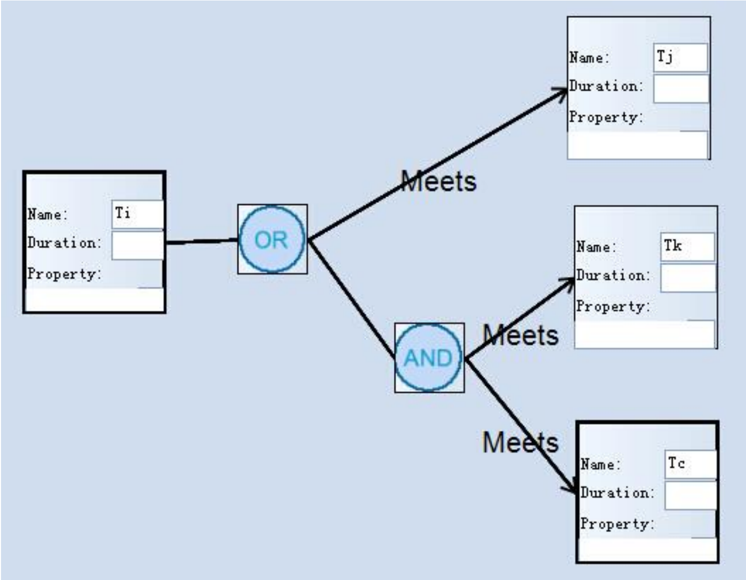


Figure 4.14 Meets(T_i,T_k)∧Meets(T_i,T_c)∨Meets(T_i,T_j)

■ Meets(T_i,T_k)∧(Meets(T_i,T_c) ∨Meets(T_i,T_j))(See Figure 4.15)

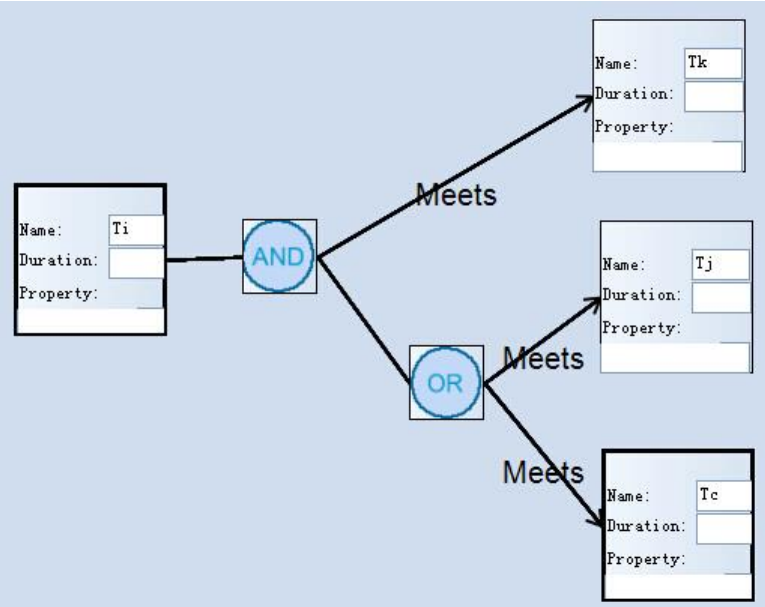


Figure 4.15 Meets(T_i,T_k)∧(Meets(T_i,T_c) ∨Meets(T_i,T_j))

- Meets(T_i,T_k)∧(Meets(T_c,T_k)∨Meets(T_j,T_k))(See Figure 4.16)

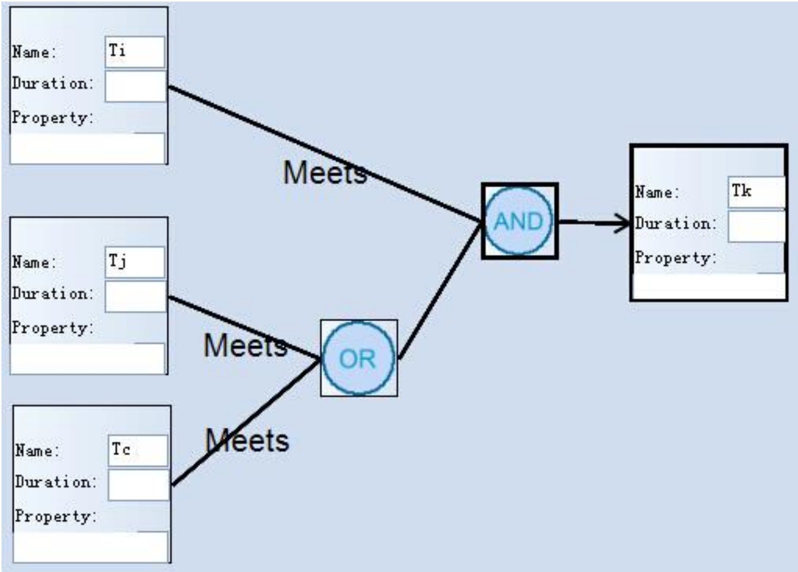


Figure 4.16 Meets(T_i,T_k)∧(Meets(T_c,T_k)∨Meets(T_j,T_k))

- (Meets(T_j,T_k)∨Meets(T_c,T_k))∧(Meets(T_k,T_a)∨Meets(T_k,T_b))(See Figure 4.17)

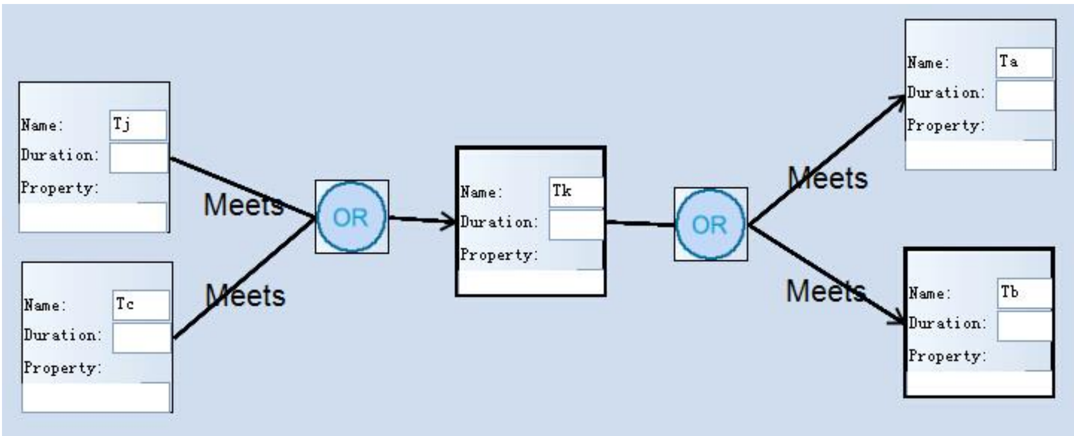


Figure 4.17 (Meets(T_i,T_k)∨Meets(T_j,T_k))∧(Meets(T_k,T_a)∨Meets(T_k,T_b))

There are two limitations of this TRD:

1 Relation(T1,T3) $\vee$ Relation(T2,T4),Then T1 = T2 $\Rightarrow$ T3 $\neq$ T4;

2 Relation(T1,T3) $\vee$ Relation(T2,T4),Then T3 = T4 $\Rightarrow$ T1 $\neq$ T2;

[Sad2012] introduces a novel temporal framework for relational event representation. However, Visual Time has two advantages, which are: 1) Visual Time supports visualizing uncertainty and incompleteness of temporal knowledge; 2) The user interface of Visual Time is friendlier.

Section 4.3.2 A TRD Matching Tool

Every single time element must have a corresponding natural language sentence(s). This TRD algorithm can be expressed by a visual TRD tool, which shows the relations between natural language information and TRD. The aims of this tool are:

1. When a natural sentence was chosen in Natural Language Description view, its related time elements should be highlighted in Time Relation Diagram view.
2. When time element(s) was chosen in Time Relation Diagram view, its related natural sentence(s) should be highlighted in Natural Language Description view

The effect of this tool is shown from Figures 4.18, to 4.21.

About 25 youths smashed their way into the warehouse, looted it and then threw two petrol bombs, leaving the building to burn for almost 10 days, resulting in its total destruction.

The above sentence was picked up from BBC news on 20 May 2014 named "*Businesses win riot damage ruling*".

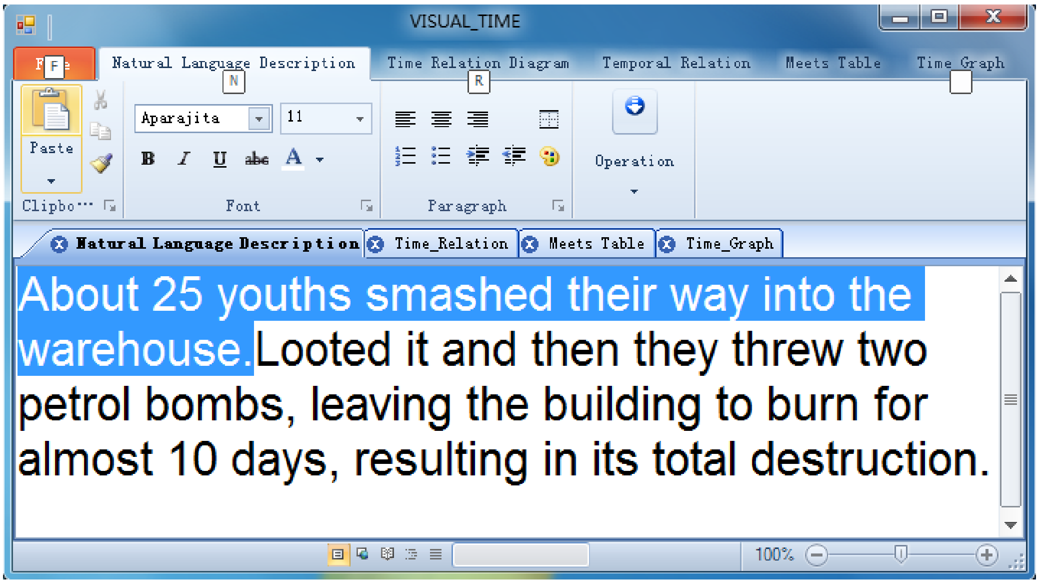


Figure 4.18 BBC Example

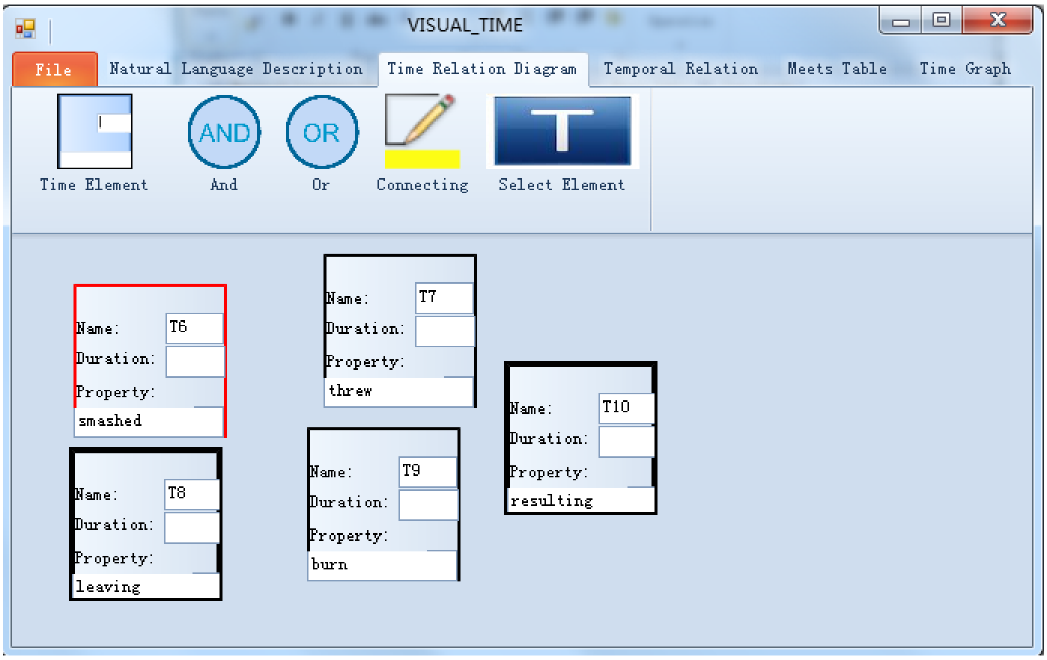


Figure 4.19 Element Select

Figure 4.20 shows a chosen natural sentence, and Figure 4.21 shows that the border colour of corresponding time element "T6" has turned to red.

In Figure 4.20, all the time elements were selected; Figure 4.21 shows that all the natural information was underlined.

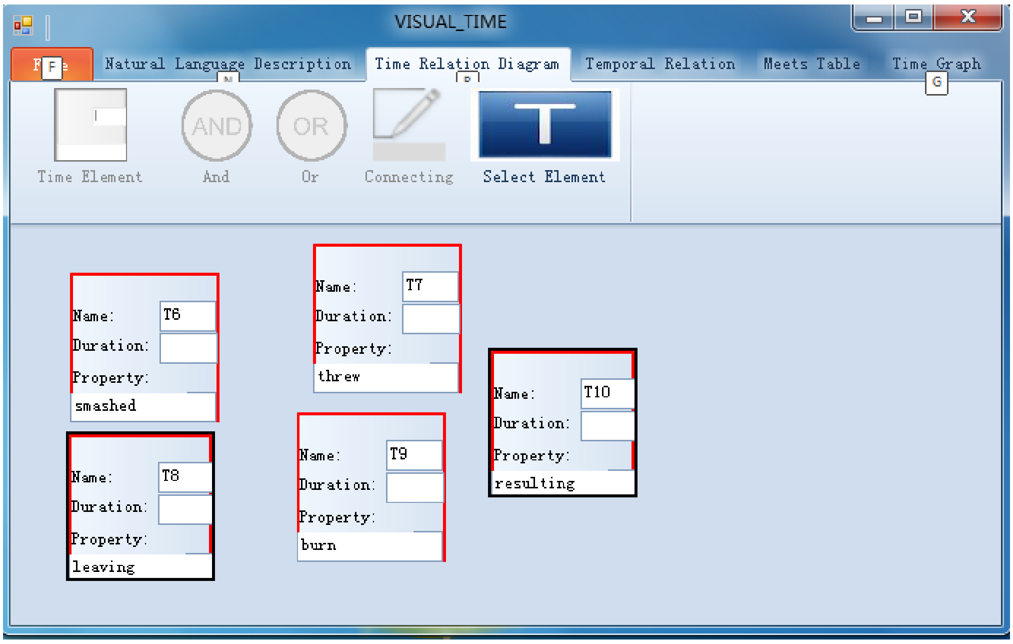


Figure 4.20 Select All Elements

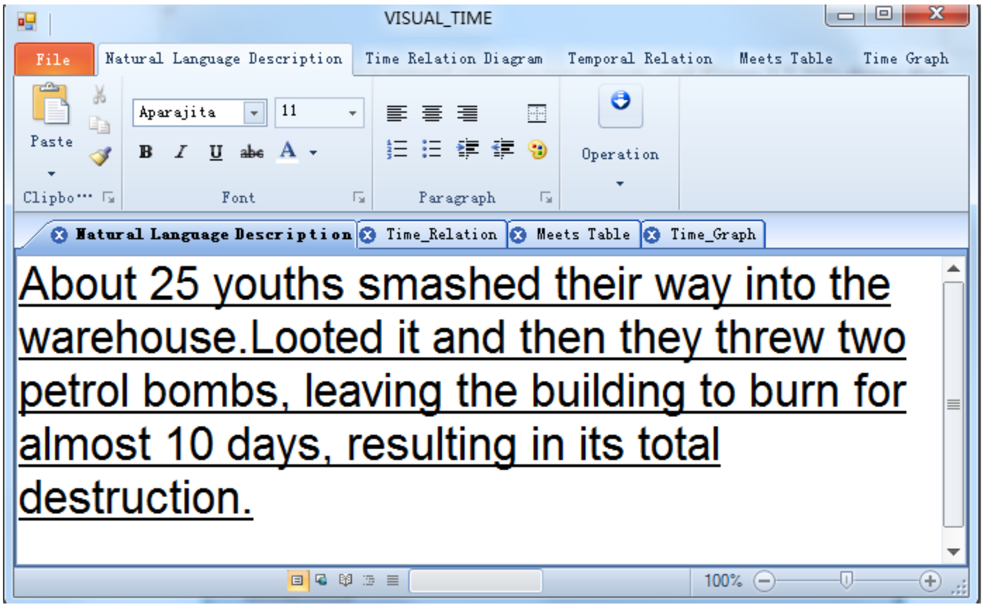


Figure 4.21 Natural Result

Section 4.3 Temporal Relation Algorithm

The temporal relation algorithm is designed to extract the corresponding temporal relations from TRD. The following pseudocode shows how to transform the TRD into Time relations:

Input: TRD;
Output: Temporal relation list (initialized as empty);
Begin
    Verify all time element(s);
    For int i = 0 to the number of time element(s);
        If time element[i] has no father element, save time element[i] in Root list;

    For int i = 0 to the length of Root list;
        Temporal_relation=**Depth-First Traversal (Root[i],** Temporal_relation_list **);**
        Save temporal_relation in Temporal relation list;
End;

Function Depth-First Traversal(C, Temporal_relation_list**)**
Input: A root time element;
Begin
    For each son element of C
        If C[i] has no son time element, then return (C.son_Temporal_relation);
        Else Temporal relation = **Depth-First Traversal(C.son);**

        If C[i] is a And node or OR node, then return(Temporal relation);
        Else Save temporal_relation in Temporal relation list;
End;

Step how to **Verify all element(s)** is explained as follows:

1) Two time elements cannot have a same name;
2) Duration of time point must be zero;
3) Duration of time interval must be positive;
4) Every time element must has temporal relation with other time elements;
5) Node And/Or must has at least one father node and two son nodes.

The output will be stored in the following form

$$Meets(T1,T2) \vee (Meets(T1,T5) \wedge Before(T1,T6)).$$

N.B. Sometimes user may draw a mixture diagram. In Figure 4.22, time elements within the red box will be transformed first and the blue box will be transformed later.

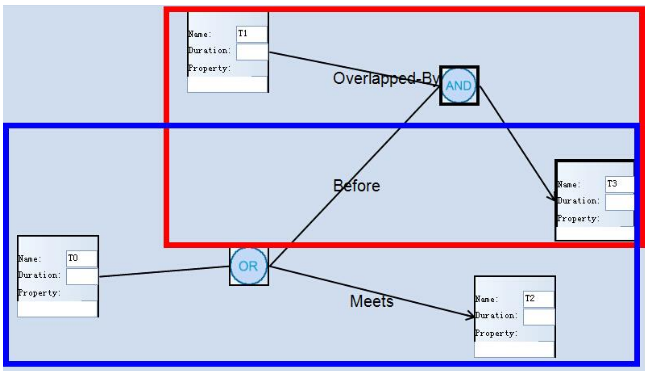


Figure 4.22 TRD Process Order

Section 4.4 Meets Table Algorithm

Meets table algorithm is designed to characterize temporal relations into a Meets table. The operator precedence is shown in Table 4.1. Table 4.2 shows how to use Meets relation to express the other temporal relations.

Table 4.1 Operator Precedence

Operator	**Description**	**Level**	**Associativity**
( )	Parentheses	1	Left-to-right
$\wedge$	Conjunction (and)	2	Left-to-right
$\vee$	Disjunction (or)	3	Left-to-right

Table 4.2 Temporal Relations

Relation	Characterized (Meets relation)
Equal(t_1, t_2)	$\exists t',t'' \in \mathbf{T}(Meets(t',t_1) \wedge Meets(t',t_2) \wedge Meets(t_1,t'') \wedge Meets(t_2, t''))$
Before(t_1, t_2)	$\exists t \in \mathbf{T}(Meets(t_1, t) \wedge Meets(t, t_2))$
Overlaps(t_1, t_2)	$\exists t,t_3,t_4 \in \mathbf{T}(t_1 = t_3 \oplus t \wedge t_2 = t \oplus t_4)$
Starts(t_1, t_2)	$\exists t \in \mathbf{T}(t_2 = t_1 \oplus t)$
During(t_1, t_2)	$\exists t_3,t_4 \in \mathbf{T}(t_2 = t_3 \oplus t_1 \oplus t_4)$
Finishes(t_1, t_2)	$\exists t \in \mathbf{T}(t_2 = t \oplus t_1)$
After(t_1, t_2)	Before(t_2, t_1)
Overlapped-by(t_1,t_2)	Overlaps(t_2, t_1)
Started-by(t_1, t_2)	Starts(t_2, t_1)
Contains(t_1, t_2)	During(t_2, t_1)
Finished-by(t_1,t_2)	Finishes(t_2,t_1)
Met-by(t_1,t_2)	Meets(t_2,t_1)

The following pseudocode shows how to transform the Time relations list into Meets Table list and how to calculate the locations of time elements in Meets table:

N.B.

1] Step **Save Meets Relation** can be explained as follows: If Meets(T1,T2), then Time element T1 will be marked as the father element of time element T2, and T2 will be marked as the son element of T1.

Input: Temporal relation list;

Output: Meets Table list (initialized as empty);

Begin

For int i = 0 to the length of Temporal relation list;

Characterize temporal relation[i] according to Table 4.2;

Save Meets Relation;

For int i = 0 to the length of Meets Table list;

If Meets Table[i] has no father element, save Meets Table[i] in Root list;

For int i = 0 to the length of Root list;

Draw_time_graph(Root[i]);

End;

Function Draw_time_graph(**D**);

Input: D;

Begin

If D.start_location = (null,null)&& D.end_location =(null,null)

D.start_location = (200 + Random(200),200);

D.end_location = (200 + Random(200),400);

If D.start_location $\neq$ (null,null)&& D.end_location =(null,null)

D.end_location = (D.start_location.x+ Random(200), D.start_location.y + Random(200));

```
If D.start_location = (null,null)&& D.end_location ≠ (null,null)
    D.start_location = (D.end_location.x- Random(200), D.end_location.y - Random(200));

For each father element of D
    D.father[k].end_location = D.start_location;
For each son element of D
    D.son[j].start_location = D.end_location;
    Draw_time_graph( D.son[j])

End;
```

To demonstrate how to use the Meets Table Algorithm an example is provided below:

<EE> Meets(T1,T2)∨(Meets(T1,T5)∧Before(T1,T6))

In Step 1), Before(T1,T6) is expressed by Meets(T1,T1'6')∧Meets(T1'6',T6) ;where <EE> can be expressed by

Meets(T1,T2)∨(Meets(T1,T5)∧Meets(T1,T1'6')∧Meets(T1'6',T6))

In step 2), according to the operator precedence shown in Table 4.1, Meets(T1,T5)∧ Meets(T1,T1'6')∧Meets(T1'6',T6) is drawn first. Then according to the representation shown in Section 3.3, Meets(T1,T2)is drawn later.(See Figure 4.23)

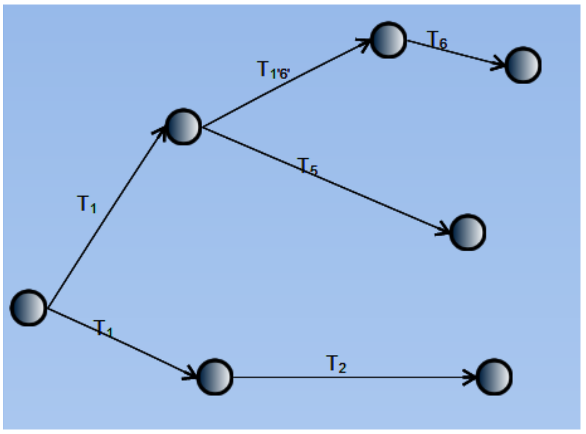


Figure 4.23 An Example For Meets Table Relations

Section 4.5 Consistency Checking Algorithm

A consistency checking algorithm is designed to check the consistency of corresponding Time graph. The Depth First Scanning Algorithm is used here. The consistency checking algorithm is as follows:

N.B.

2] **temporal_element.path** contains the corresponding time element from root element to temporal _element;

3] **temporal_element.path_duration** is the summation of the corresponding time duration(s) of time element(s) from root element to temporal _element;

Input: Meets Table list;

Output: inconsistent_time_element list (initialized as empty);

Begin

If Meets Table.length < 2 then **End;**

For i = 0 to Meets_Table.length-1

If Meets_Table[i]. temporal_relation is null, then delete Meets_Table[i];

If Meets_Table[i] has no father time element, then add Meets_Table[i] to root list;

For i = 0 to root.length-1

temporal_element = root[i];

temporal_element.path = root[i];

temporal_element.path_duration = root[i]. duration;

Depth-First Traversal(temporal _element, inconsistent_time_element list);

End;

Function Depth-First Traversal(temporal _element, inconsistent_time_element list)

Input: temporal_element

Begin

For each temporal_element's son

If temporal_element.son.path $\neq$ null;

If temporal_element.son.path_duration $\neq$ temporal_element.path_duration + temporal_element.son.duration

Record temporal_element.son.path and temporal_element.path + temporal_element.son into inconsistent_time_element list;

Else temporal_element.son.path_duration = temporal_element.path_duration + temporal_element.son.duration;

temporal_element.son.path = temporal_element.path + temporal_element.son;

Depth-First Traversal(temporal _element.son, inconsistent_time_element list);

End;

The output of this consistency checking algorithm is a inconsistent_time_element list which includes every time element that makes TRD inconsistent. According to this inconsistent_time_element list, all the corresponding time element in TRD and the corresponding natural sentence(s) will be highlighted.

Section 4.6 Conclusions

This chapter presents the Visual Time, for representing and reasoning about incomplete and uncertain temporal information. It is both expressive and versatile, allowing logical conjunctions and disjunctions of both absolute and relative temporal relations, such as “Before”, “Meets”, “Overlaps”, “Starts”, “During”, and “Finishes”, etc.

In terms of a visualized framework, Visual Time provides a user-friendly environment for describing scenarios with rich temporal structure in natural language, which can be formatted as structured temporal phrases and modelled in terms of Time Relationship Diagrams (TRD). A time relation diagram (TRD) is designed for representing temporal relations between time elements, which could be both point and interval. Each time element is denoted as a box consisting of three components: Name, Duration, and Property. Temporal relations are denoted in terms of directed arcs. TRD allows expressions of both absolute and relative temporal relations, supporting both logical conjunctions and disjunctions.

Four algorithms are introduced: The first algorithm, Temporal Relation Algorithm (TRM), is designed to extract temporal relations from TRD. The second algorithm, Meets Table Algorithm (MTM) is introduced to convert all the extracted temporal relations into a Meets table. The third algorithm, Time Graph Algorithm (TGM) is described to draw the corresponding time graph of a given TRD. The fourth algorithm, Consistency Checking Algorithm (CCM), is designed to check the consistency of TRD. If the TRD is inconsistent, an audio verdict will alert and the corresponding time element(s) and natural texts will be marked in red colour.

Some observations on the algorithms introduced in this chapter:

1 The semi-automatic temporal information extractor has never been tested over large volume of natural text;

2 When the number of time elements in TRD increases, the reaction of Visual Time becomes very slow;

3 One limitation of the Consistency Checking Algorithm is the scanning speed becomes very low when the number of time elements increases.

CHAPTER 5 A SCENARIO

The framework proposed in this thesis, Visual Time, is a computer based visualized case tool for dealing with temporal information. In what follows in this chapter, we shall use a virtual terrorist attach scenario to demonstrate the whole process of applying Visual Time to real life applications.

Section 5.1 Introduction to the Scenarios

Consider the scenario that a virtual terrorist attack happened at the University of Greenwich. After the attack happened, an enormous number of pictures and texts about the attack spread explosively over internet. In particular, social media users sent a large number of tweets on Twitter.

Visual Time can be used to collect all the tweets including texts and pictures sent within a chosen geographical area. The Scotland Yard can use the Visual Time to find hot users around the University of Greenwich who may be the potential witness. Police officers can collected all the tweets to locate suspects; after the suspects are captured, Visual Time can be used to build a corresponding TRD of suspects' statement by using the semi-automatic temporal relation extractor; the time relation algorithm can help to convert the TRD into a corresponding Time relation Table; then, the Meets table algorithm can be used to characterize the Time relation table into a

Meets table. After that, by using the extended time graph theory which is introduced in Chapter 3, a time graph will be drawn in the Time Graph View. Finally, a consistency checker can be used to check the consistency of suspects' statements.

Section 5.2 Filtering social media users

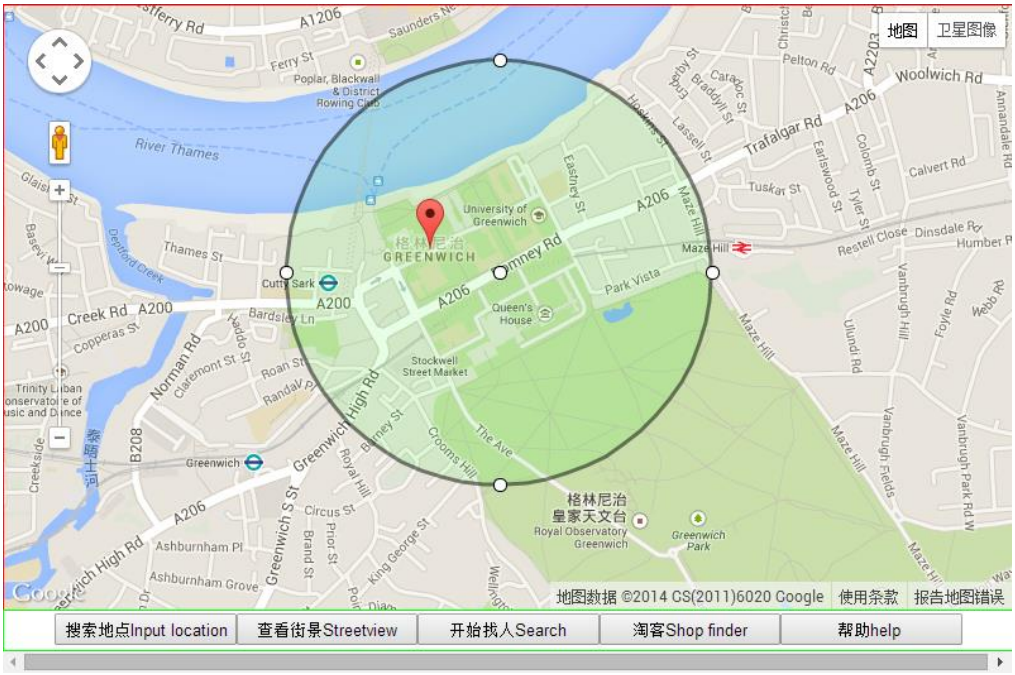


Figure 5.1 GPS Locating

In Figure 5.1, a cycle is drawn by a police officer. Visual Time can help police officers to collect all the social media users who have ever sent a tweet recently. After being filtered by Visual Time, a potential users list will be shown (See Figure 5.2).

Figure 5.2 Potential Witness

In Figure 5.2, the coordinate (51.47, 0.00) is the centre of chosen geographical area, while 1208.20 meters is the radius of the drawn cycle. Visual Time also supports police officers sending private message to the filtered hot users by setting σ-standard as zero. The texts or pictures sent on social media will be stored in the format of Table 5.1.

Table 5.1 Format of Social Media Data

id_str	467212189120884736
from_user	Tom
text	I may see the suspects.
created_at	Fri May 16 07:57:20 +0000 2014
time	5-16-2014 8:57:20
geo_coordinates	loc: 51.48185306,-0.00784056
iso_language_code	
to_user_id_str	
from_user_id_str	15078144
source	Instagram
status_url	http://twitter.com/Puplett/statuses/467212189120884736

The security experts can use street view to check the geographical environment, such as road layout and building location, on their iPad or Laptop when they are on their way to the university (See Figure 5.3). With this information, security experts may use CCTV images from the involved locations to track the potential attackers.

Figure 5.3 Street View of Greenwich Campus

After critical analysis and investigations, Scotland Yard finally captured three suspects Peter, Jack and Robert. The police officers can use Visual Time to record the statements given by the suspects (see next section).

Section 5.3 TRD Generating

As the suspects Jack, Peter and Robert gave the following statements, respectively in Figure 5.4:

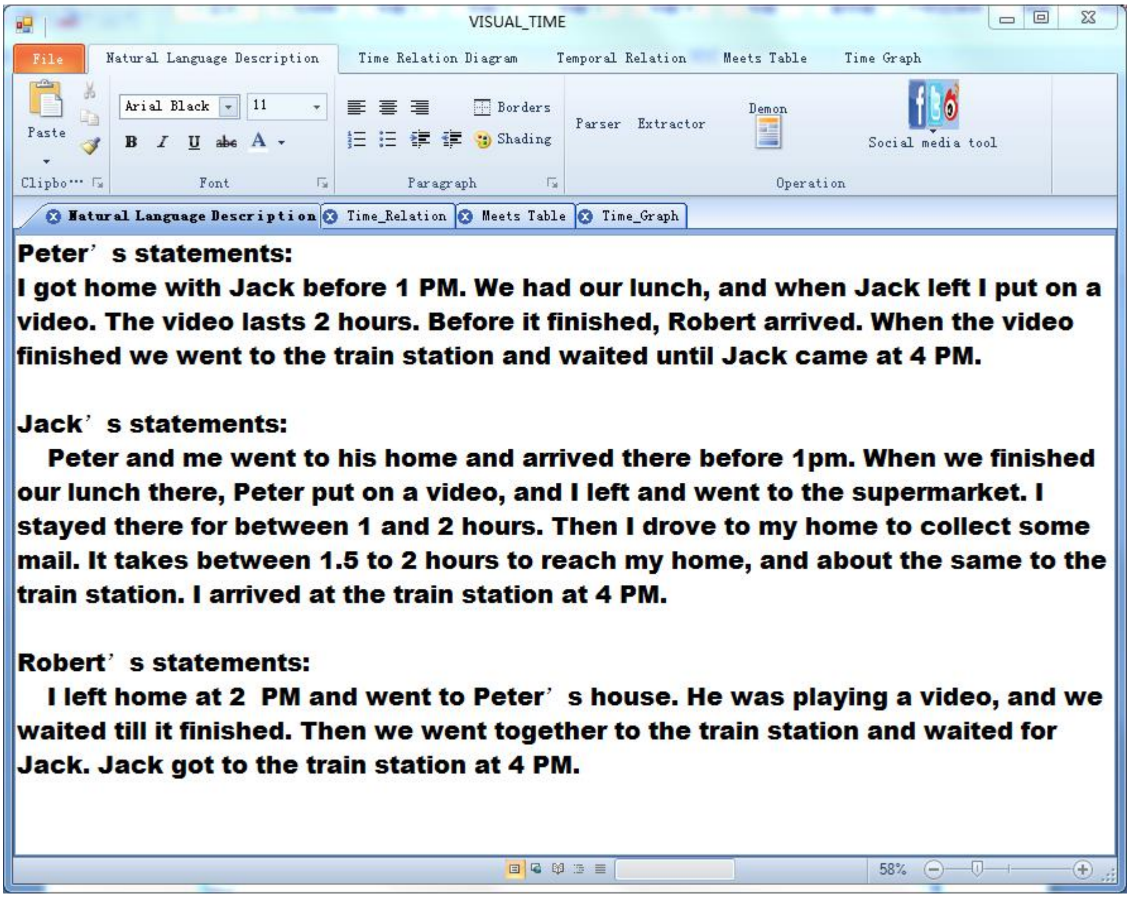


Figure 5.4 Natural Language Description

According to the given natural language description, Visual Time can semi-automatically extract all the temporal information, which will be used to draw a TRD (See Figure 5.5).

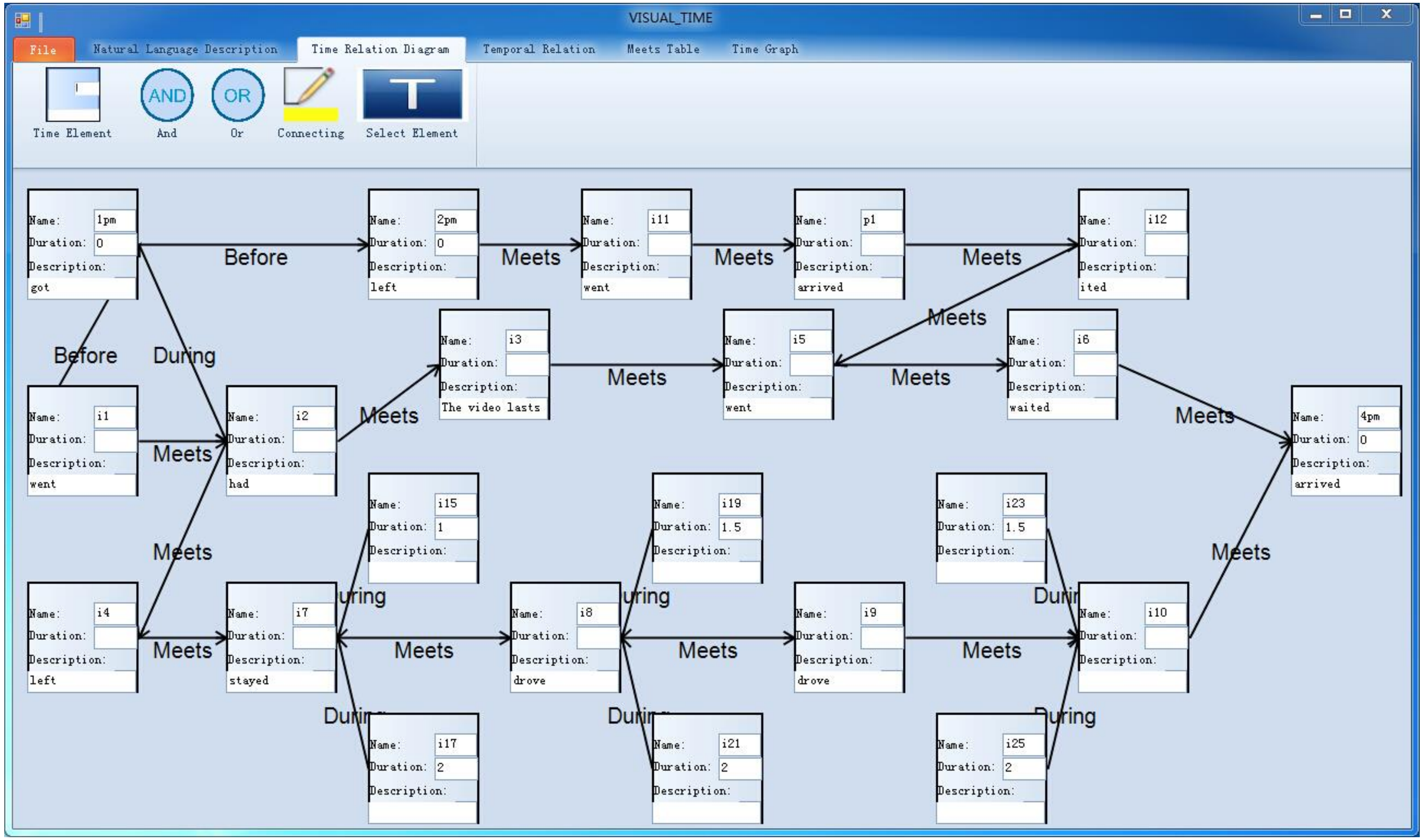


Figure 5.5 TRD of given scenario

Each single element in Figure 5.5 has its corresponding verb in Figure 5.4.

Section 5.4 Temporal Relation

The temporal relation algorithm will extract all the temporal relations from TRD into its corresponding time relation table, which is shown in next section. This can be done by simply pressing the “Temporal Relation” button in Visual Time.

Figure 5.6 shows the corresponding time relation table of the TRD.

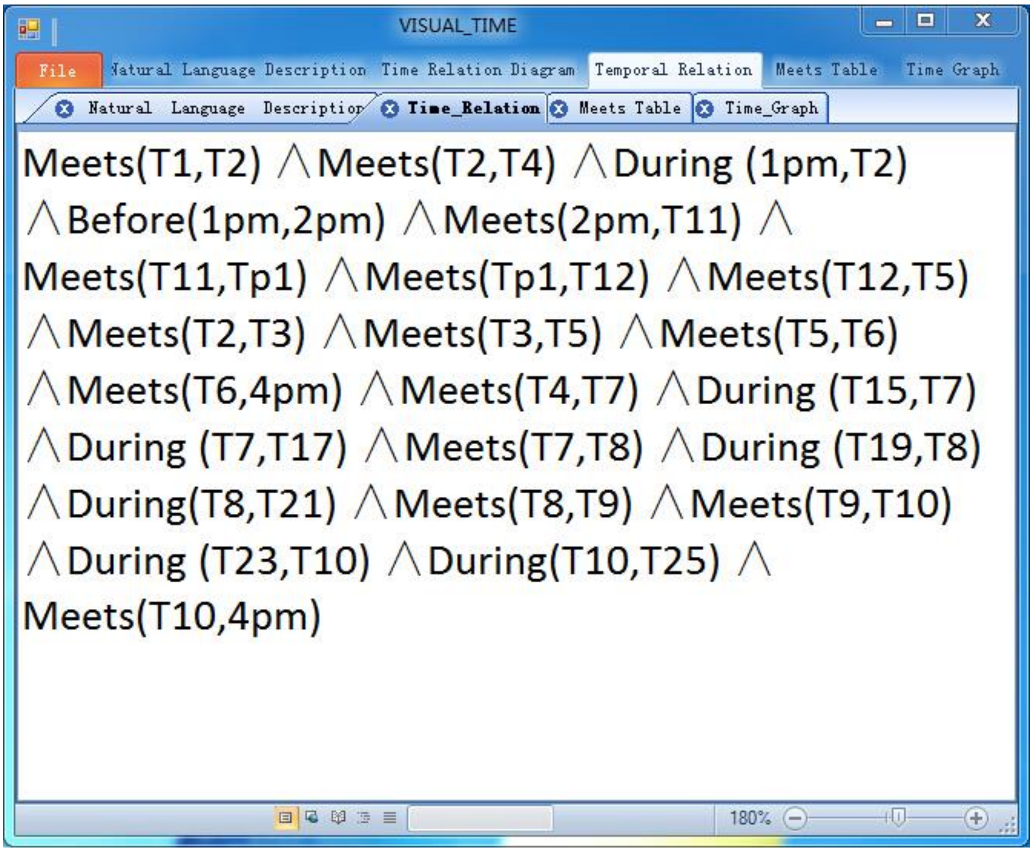


Figure 5.6 Time Relation

Section 5.5 Meets Table

As illustrated in Table 4.1 of Section 4.4, each of the 13 temporal relations can be expressed by a conjunction of Meets relations. For example:

$$Before(t_1, t_2) = \exists t(Meets(t_1, t) \wedge Meets(t, t_2))$$

Based on the temporal relations shown in Figure 5.6, the Meets table algorithm designed for Visual Time can be used to translate the temporal relations in Figure 5.6 into a Meets table. This can be done by simply pressing the “Meets Table” button in Visual Time.

Figure 5.7 shows the corresponding Meets table of the given scenario.

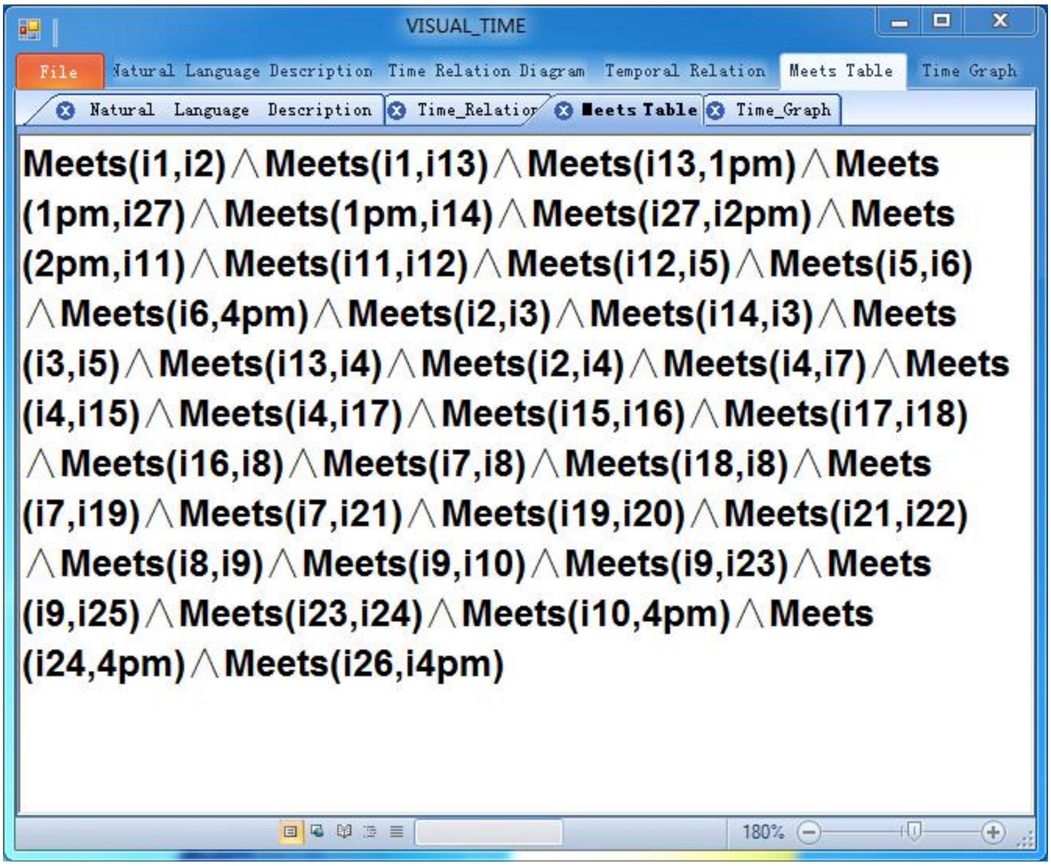


Figure 5.7 Meets Table

Section 5.6 Graphically Representing

Based on the temporal relations shown in Figure 5.8, Visual Time can be used to draw the corresponding Time graph of the Meets table in Figure 5.8. This can be done by simply pressing the “Time Graph” button in Visual Time.

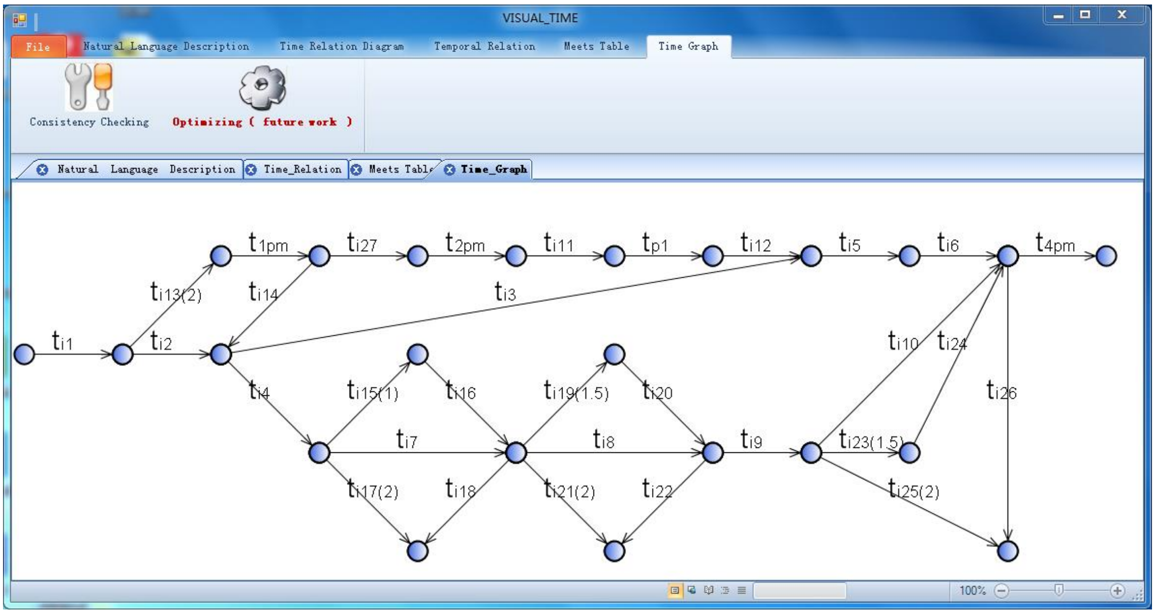


Figure 5.8 Graphically Representing

Figure 5.8 shows the corresponding Time graph of given scenario.

N.B. the absolute time elements 1:00 PM, 2:00 PM and 4:00 PM are expressed by t_{1pm}, t_{2pm} and t_{4pm} respectively.

In next section, a consistency algorithm will be used to check the consistency of given scenario.

Section 5.7 Consistency Checking

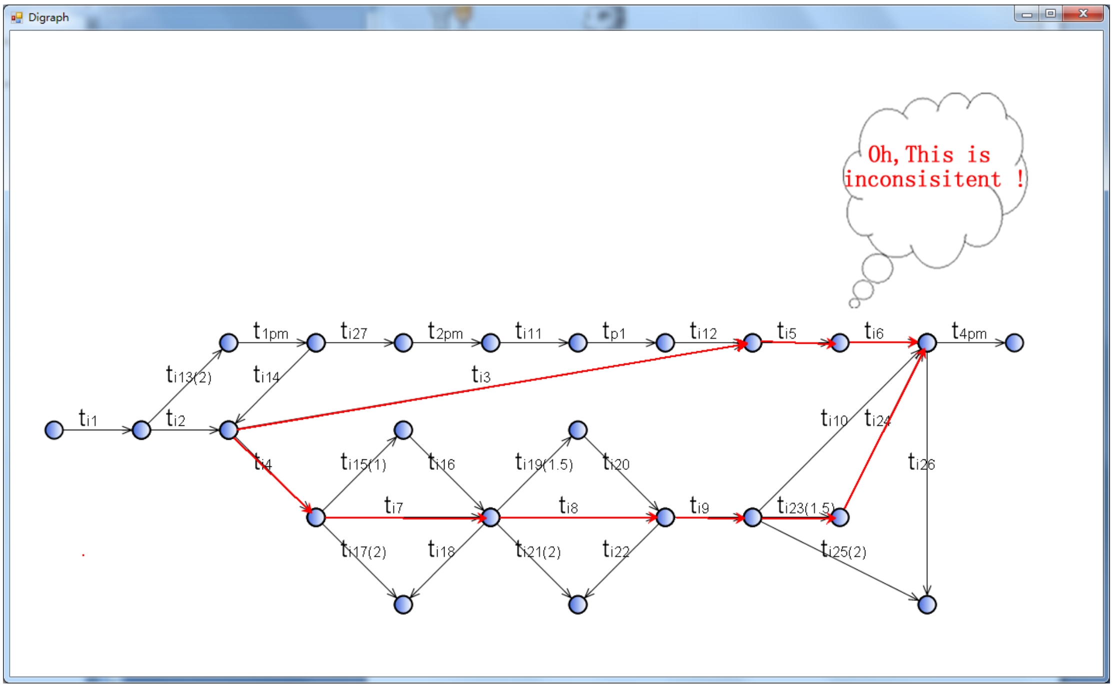


Figure 5.9 Checking Result

In Figure 5.9, the consistency result shows the given scenario is inconsistent.

Since each interval has a positive duration; each point has a non-negative duration; t_5 is before t_{1pm}; t_6 Meets t_{4pm}; 4:00 PM is 2 hours before 2:00 PM. This leads to the following result:

$$Dur(T_5) + Dur(T_6) < 2$$

In addition, since

$$Dur(T_3) = 2$$

Hence:

$$Dur(T_3) + Dur(T_5) + Dur(T_6) < 2 + 2 = 4$$

However,

$$Dur(T_4) + Dur(T_7) + Dur(T_8) + Dur(T_9) + Dur(T_{23}) + Dur(T_{24})$$

$$> 0 + 1 + 1.5 + 0 + 1.5 = 4$$

The strict inequalities appearing in each of the above two cases show that:

On one hand, $Dur(T_3) + Dur(T_5) + Dur(T_6)$ is less than 4, and on the other hand, $Dur(T_4) + Dur(T_7) + Dur(T_8) + Dur(T_9) + Dur(T_{23}) + Dur(T_{24})$ is bigger than 4. Therefore,

$$Dur(T_3) + Dur(T_5) + Dur(T_6)$$

$$\neq Dur(T_4) + Dur(T_7) + Dur(T_8) + Dur(T_9) + Dur(T_{23}) + Dur(T_{24})$$

that is,

$Dur(T_3) + Dur(T_5) + Dur(T_6) - Dur(T_4) - Dur(T_7) - Dur(T_8) - Dur(T_9) - Dur(T_{23}) - Dur(T_{24}) \neq 0$

Hence, the time graph shown in Figure 5.9 is inconsistent.

Figure 5.9 shows that the given scenario is inconsistent. The time elements, which make it inconsistent, are T_3, T_5, T_6, T_4, T_7, T_8, T_9, T_{23} and T_{24}. As mentioned in Section 4.3.2, every the time element links a natural sentence which contains the time/action in the natural language description.

In Figure 5.10, the colours of related sentences, which make this natural information inconsistent, have been turned to red. Therefore, it is very easy to get the checking results in Figure 5.10 that Peter and jack statements are inconsistence.

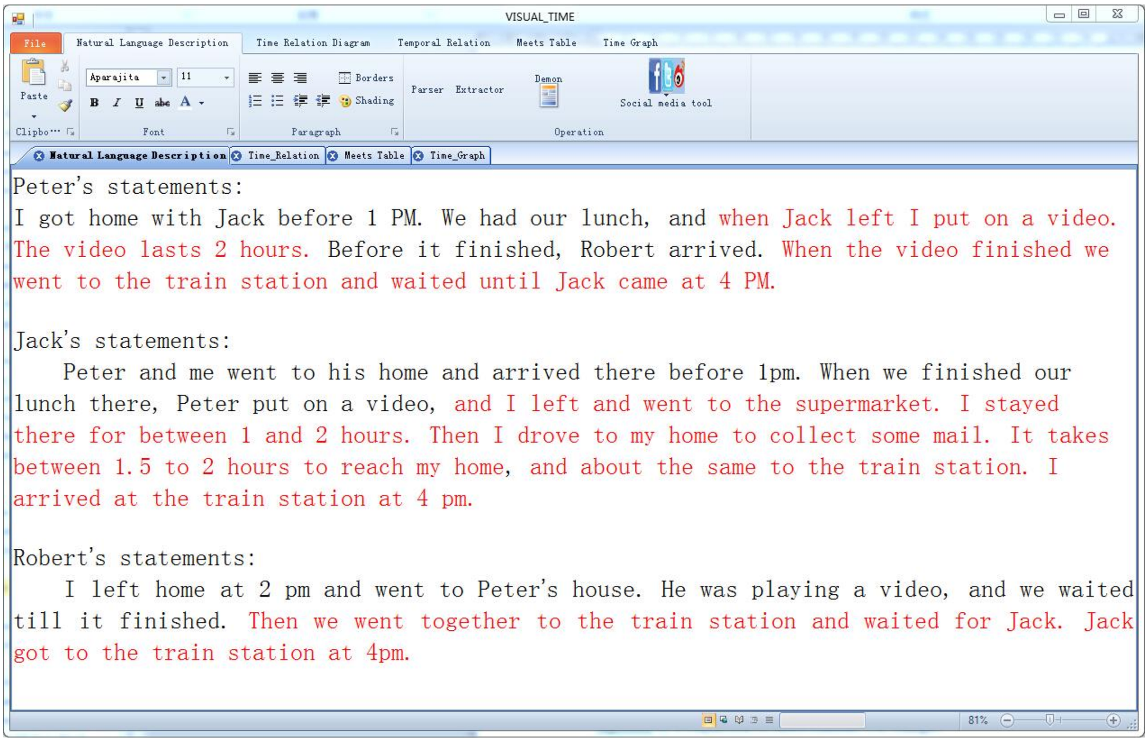


Figure 5.10 Natural Checking Results

Section 5.8 Conclusion

This scenario uses a virtual scenario with a background of a terrorist attack happening in the University of Greenwich to show how Visual Time helps in public security affairs. Security experts can use it to dynamically track the information on the internet and also can filter the social media users to find witness among the hot users. Then the TRD will be generated from suspects' natural statements. An automatic consistency checker will help to check the consistency of TRD, and the checking result will be shown in the original natural language description.

This framework not only can be used in public security affairs, but also can be used for business planning, automatic question answering and decision making, etc. Take another example, if a company wants to check the cost of time of their product, they can use this framework to check the plan for consistency.

One limitation of this framework is that Visual Time cannot find the best results of the given scenario. This thesis will not spend time to research these kinds of situations, but this problem will be discussed in the future work section.

CHAPTER 6 SUMMARY CONCLUSUION

To conclude, the main idea of this thesis is to present a visualized framework for uncertain or incomplete temporal knowledge, called Visual Time, for filtering big data among social media websites, representing and reasoning about incomplete and uncertain temporal natural information. In terms of a visualized framework, Visual Time provides a user-friendly environment for describing scenarios with rich temporal structure in natural language, which can be automatically formatted as structured temporal phrases and modelled in terms of Temporal Relationship Diagrams (TRD). TRD can be automatically and visually transformed into a matching Time Graph, supported by automatic consistency checker that derives a verdict to confirm if a given scenario is temporally consistent or inconsistent. And the consistency result of the scenario will be graphically shown in the natural language.

Section 6.1 Contribution Restate

This thesis contributes to the area of Artificial Intelligence. A visualized framework for uncertain and incomplete temporal knowledge called Visual Time is proposed. Four main contributions are summarized as follows:

1) **Extended graphical representation for uncertain and incomplete temporal knowledge:** An extended graphical representation for uncertain and incomplete temporal knowledge based on [KM1992] is proposed, supporting both logical connectives ‘$\wedge$’ and ‘$\vee$’. In Chapter 3, it is shown all the other logical connectives can be derively defined.

2) **Time relation diagram (TRD):** A time relation diagram (TRD) is designed for representing temporal relations between time elements which could be both point and interval. Each time element is denoted as a box consisting of three components: Name, Duration and Property. Temporal relations are denoted in terms of directed arcs. TRD allows expressions of both absolute and relative temporal relations, supporting both logical conjunctions and disjunctions.

3) **A semi-automatic temporal information extractor:** SUTime is a very useful tool for extracting verbs and temporal information [CM2012]. However, the extracted verbs and temporal information may play different roles when modelled by TRD. For example, in "*He starts to start the car*", "*start*" is an event while "*starts*" means the action "*start*" happens. An improved algorithm called Temporal Extractor algorithm (TE) is introduced in Section 4.2. Based on Stanford SUTime, TE can semi-automatically extract time elements and temporal relations from any arbitrary text to create a TRD.

4) Four algorithms: The first algorithm, Temporal Relation Algorithm (TRM), is designed to extract temporal relations from TRD. The second algorithm, Meets Table Algorithm (MTM) is introduced to convert all the extracted temporal relations into a Meets table. The third algorithm, Time Graph Algorithm (TGM) is described to draw the corresponding time graph of a given TRD. The fourth algorithm, Consistency Checking Algorithm (CCM), is designed to check the consistency of TRD. If the TRD is inconsistent, an audio verdict will alert and the corresponding time element(s) and natural texts will be marked in red colour.

All in all, it is believed that visual time is a very practical tool for filtering big data among social media websites, representing and reasoning about incomplete and uncertain temporal natural information. Though the proposed systems have not been actually used by some companies or governments, we are confident that they will draw more scientists' attention and provide valuable references to them.

Section 6.2 The Difference between Current and Previous Work

1. **Extended graphical representation for uncertain and incomplete temporal knowledge:**

   There are two differences between this thesis and previous work:

   i. Previous work [KM1992] and [JBMX2010] have not discussed graphical representation of time graph on the logical operation such as: associativity, commutativity, distributivity, idempotence, absorption, monotonicity, affinity, duality, truth-preserving, falsehood-preserving, involutivity.

   ii. Previous work [KM1992] and [JBMX2010] cannot express all the possible temporal knowledge such as: $Meets(T_i,T_k)\wedge(Meets(T_i,T_c) \vee Meets(T_i,T_j))$, $Meets(T_i,T_k)\wedge (Meets(T_c,T_k)\vee Meets(T_j,T_k))$, $(Meets(T_i,T_k)\vee Meets(T_j,T_k))\wedge (Meets(T_k,T_a)\vee Meets(T_k,T_b))$;

2. **Time relation diagram (TRD):**

This thesis is the first work which puts forward the TRD.

1. **A semi-automatic natural information extractor:** SUTime is a very useful tool for extracting verbs and temporal information [CM2012]. However, the extracted verbs and temporal information may play different roles when modelled by TRD. For example, in "*He starts to start the car*", "*start*" is an event while "*starts*" means the action "*start*" happens. An improved algorithm called Temporal Extractor algorithm (TE) is introduced in Section 4.2. Based on Stanford SUTime, TE can semi-automatically extract the time elements from any arbitrary text to create a TRD.

2. **Four TRD algorithms:** The first algorithm, Temporal Relation Algorithm (TRM), is designed to extract temporal relations from TRD. Secondly, a Meets Table Algorithm (MTM) is introduced to convert TRD into a Meets table. Thirdly, a Time Graph Algorithm (TGM) is described to draw time graph of TRD. The fourth algorithm, Consistency Checking Algorithm (CCM), is designed to check the consistency of TRD. If the TRD is inconsistent, an audio verdict will alert and the corresponding time element(s) and natural texts which will mark the corresponding scenario inconsistent in red colour.

Section 6.3 Critical Assessment Of Own Work

A temporal interval is often visually displayed by a horizontal bar on a bi-dimensional space, where the x-axis represents the time line and the y-axis is associated with the considered time-varying information.

This thesis uses a new graphical-based user interface instead of rule-based interface to express temporal intervals, which is shown in Section 2.2.1. In this way, it is quite intuitive to graphically represent any set of intervals if their respective relations belong to 13 temporal relations put forward by Allen.

There are still some limitations and shortcomings in some model of Visual Time.

1. **Social media searching model:** Social media searching model supports users drawing a circle over any area of the Google map to filter social media users, which support both English and Chinese. There are a lot of open source projects, which can collect social media data, but they only can support the users inputting the keywords or choosing the city name to filter data. For example, if users want to search the social media users around Greenwich Park, users just need to input the keyword "Greenwich Park, London". But if users want to search the social media users within south UK or between London and Manchester, no social media searching tool supports these features. However, there are still some problems in Social media searching model.

    - Only twitter and sina weibo were supported. User cannot use it on Facebook or some other social media website. Because of the time restriction, this thesis had no enough time to develop every social media company.
    - No flexible time search model was developed. For example, user cannot search what happened 7 days ago. This model only supports searching the temporal data happening during the last 7 days.

2. **Natural language Extractor model:** Stanford University is the leading research group on this aspect. Based on their work (SUTime and Parser), this thesis developed an improved NLP extractor which can help people extract most of the time and actions to model in TRD. The problems of this natural language extractor are as follows:

- Time relation extract: This framework cannot recognize all of Allen's 13 temporal relations from natural language information. Users will have to draw the temporal relation manually when the natural information includes some complicated texts.

- Emotion extract: This framework can extract most of time and actions among natural language knowledge, but up to now, very few words can be understood. The framework cannot understand some natural information like "*I got home before 1:00 PM, and I feel tired.*" The emotion of natural information "*tired*" will be ignored by this framework. However, some scientists insist that emotion is very important for predicting the future action of people.

- Lack of natural *Meets* relation: Since all of Allen's 13 temporal relations can be characterized in a Meets table. It is quite important to create a natural language *Meets* relation such as " as soon as possible", "immediately" and so on.

CHAPTER 7 FUTURE WORKS

There is room for the improvement and generalization of the framework. In this chapter, further work to Natural language Extracting, Consistency Improvement, Generalized Uncertain or Incomplete Temporal Relations and Public Security are discussed.

Section 7.1 Improved Natural language Extracting

This framework can extract all time elements and temporal relations - before and after. However, to extract the other temporal relations, some further works is required:

Improvement on temporal relation extracting: Some of these relations like *overlaps, during* is still impossible to extract by this framework. Take the example described in Section 2.1.5, it is possible to express the situation that “X was born before Y’s death” can be expressed as “X lived before Y or X’ life meets Y’s life or X’ life overlaps Y’s life or X’ life is finished by Y’s life or X’ life contains Y’s life or X’ life is started by Y’s life or X’ life equals Y’s life or X’ life starts Y’s life or X lived during Y’s life or X finishes Y’s life or X’ life is overlapped by Y’s life”, and “X died after Y” can be expressed as “X’ life contains Y’s life or X’ life is started by Y’s life or X’ life is overlapped by Y’s life or X’ life is started by Y’s life or X lived after Y”. The inference step then consists of forming the conjunction of the two sets of disjunctions: “X’ life contains Y’s life or X’ life is started by Y’s life or X’ life is overlapped by Y’s life” which is equivalent to the conclusion derived above [Chr1992].

Therefore, an algorithm will be needed for temporal relations extracting.

"Meets" relation extracting: All Allen's 13 temporal relations can be expressed in terms of the Meets relation. However, in natural language, such "Meets" relations may be expressed in various forms. For example, "John called Jack as soon as he got home", etc. Further work needs to be done about how to extract "Meets" relations from natural text.

Duration extracting: The duration of time element is a key feature for modelling and consistency checking. How to assign the time length to the corresponding extracted time element is a big challenge in NLP research field.

Section 7.2 Theoretical Extension

Since this framework can get the consistency result, two questions will come as follows:

1. If TRD is inconsistent, how to make TRD consistent?

Take business planning for example, if the plan is inconsistent, the boss must want to pay the lowest price to make it consistent. Some algorithms will be considered according to the company's capacity, yield or financial conditions.

2. If TRD is consistent, can we improve TRD?

In the problem of path planning in navigation of robot, Ant Colony Optimization (ACO) [DB2005] can be used for path planning. In path optimizing and path smoothness, researchers try their best to find the shortest and cheapest path. Since Visual Time can check the consistency of given TRD, in the temporal path planning problem, further works on Visual Time are required to find the shortest and cheapest path of corresponding time graph.

Section 7.2.1 Generalized Uncertain Or Incomplete Temporal Relations

In Section 3.4, seven kinds of basic graph structures have been introduced. For logical connective OR, this thesis only considers the case such that:

1 (Relation(T1,T3) $\vee$ Relation(T2,T4)) $\Rightarrow$ (T1 = T2 $\Rightarrow$ T3 $\neq$ T4);

2 (Relation(T1,T3) $\vee$ Relation(T2,T4)) $\Rightarrow$ (T3 = T4 $\Rightarrow$ T1 $\neq$ T2);

Other cases involving logical OR that don’t satisfy the above constrains remain as future work.

1. Extended "*MEETS"* relations example:

$$Meets(T1,T2) \vee Meets(T2,T1)$$

is a uncertain temporal structure which "branching" from the different time elements (See Figure 7.1). Based on this, quite a lot of other structures will be found.

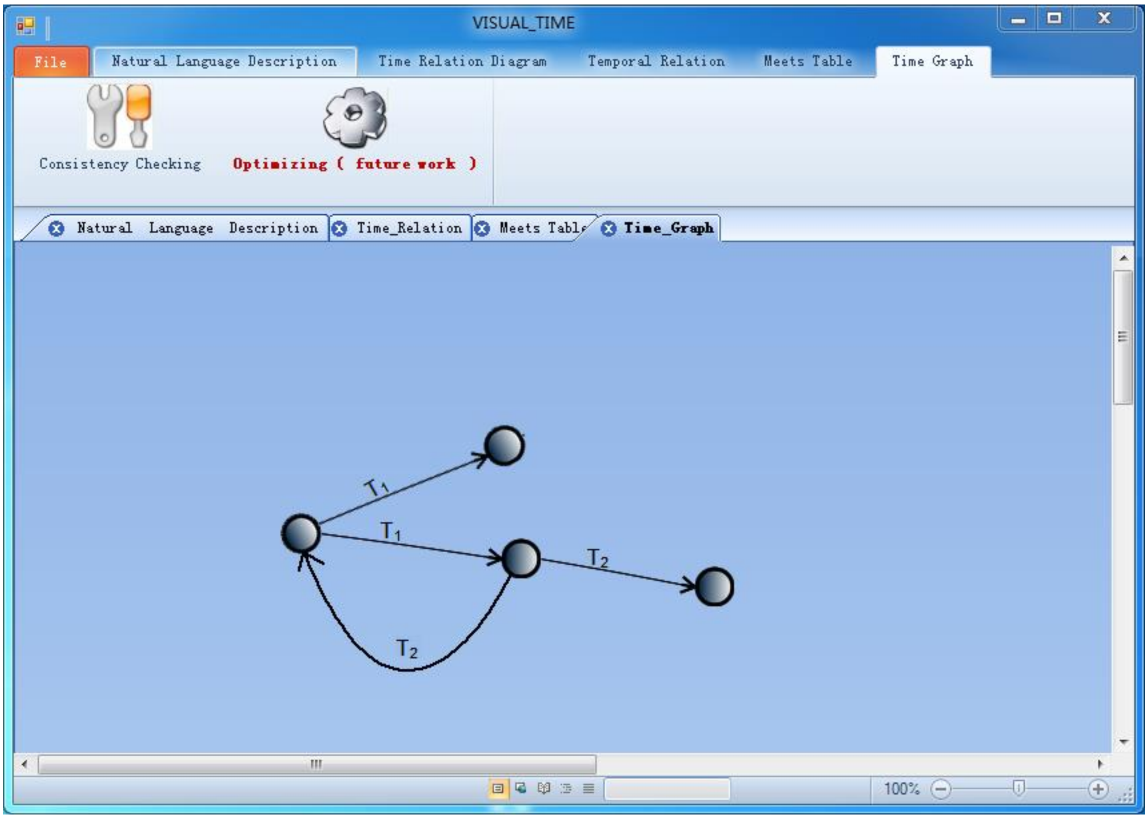


Figure 7.1 Meets(T1,T2) ∨ Meets(T2,T1)

Meets(T1,T2) ∨ Meets(T2,T1) defines that T1 Meets T2 OR T1 May Met-by T2; (See Figure 7.1)

2. Based on situation 1, one extended "*MEETS*" structure example can be given as follow:

Meets(T1,T2) ∨Meets(T3,T4)

However, *Meets(T1,T2) ∨Meets(T3,T4)* only can be graphically represented with some other structures.

3. Extended temporal relations example:

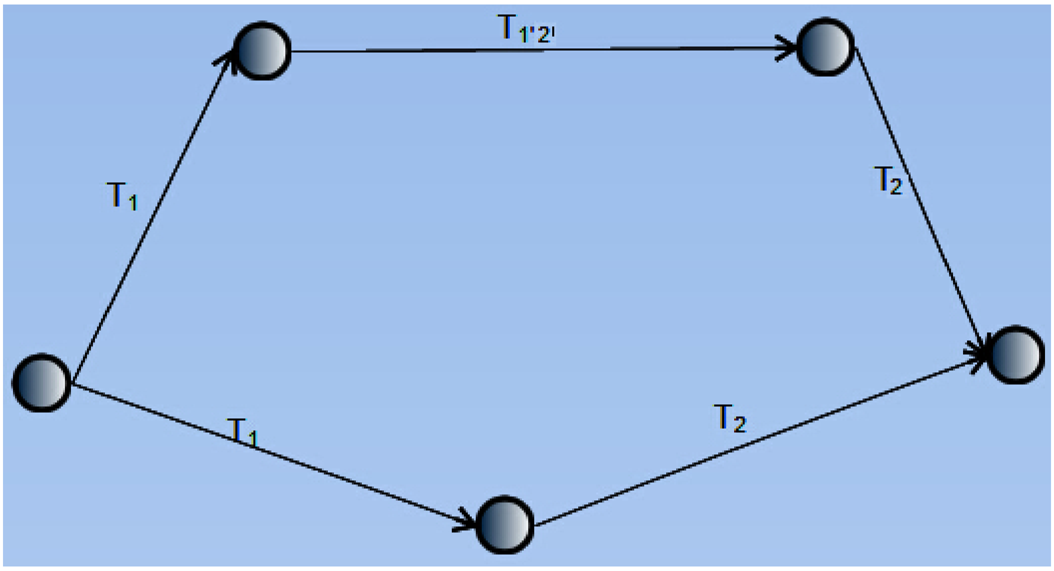


Figure 7.2 Extended Temporal Relations Example

Meets(T1,T2) $\vee$ *Before(T1,T2)* is a mixture of temporal relations which can be expressed in Figure 7.2. Starts(T1,T2) $\vee$Before(T1,T2) cannot be graphically expressed by the framework Visual Time

$$Starts(T1,T2) \vee Before(T1,T2)$$

4. Finally, an extended temporal relations example will be given as follow:

$$Meets(T1,T2) \vee Before(T3,T4)$$

However, Visual Time cannot express the above temporal relations.

Section 7.3 Public Security

The future works on temporal analysis and live CCTV can help with the public security field.

1. Temporal analysis

Los Angeles and Santa Cruz Police Departments used big data to predict crime [Mcn2011]. There has been a 33% reduction in burglaries, 21% reduction in violent crimes, and 12% reduction in property crime in the areas where predictive software is being used. However, a social media engine has been introduced in Section 4.2.1. This engine can collect a huge amount of data from internet. To extract and analyse the temporal information from the collected data can be extended into the big data research field.

2. Live CCTV

Due to the security reasons, it is legally difficult to connect to the CCTV systems of London. However, when a terrorist attack happens, by intergrading the social media and the live CCTV, security experts can use Visual Time to dynamically track the information on the internet and also can filter the social media users to find witness among the hot users.

REFERENCES

[ABH2008] Ahlqvist, Toni., Bäck, A., Halonen, M. and Heinonen, S (2008). "Social media road maps exploring the futures triggered by social media". VTT Tiedotteita – Valtion Teknillinen Tutkimuskeskus (2454): 13. Retrieved 9 December, 2012.

[AC2008] Anna Rafferty and Christopher D. Manning. Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines. In ACL Workshop on Parsing German. 2008.

[ADO1985] Adobe Systems, Inc. Postscript Language Reference Manual. Addison-Wesley, 1985.

[AF1994] Allen J. and Ferguson G. Actions and Events in Interval Temporal Logic, the Journal of Logic and Computation, 4(5), pp.531-579, 1994.

[AH1985] James F. Allen and Patrick J. Hayes.A common-sense theory of time. In Proceedings of the 9th international joint conference on Artificial intelligence, volume 1 of IJCAI'85, pages 528–531, San Francisco, CA, USA, 1985. Morgan Kaufmann Publishers Inc.

[AH1989] Allen, J. and Hayes, J., Moments and Points in an Interval-based Temporal-based Logic, Computational Intelligence, 5(4), 225-238.1989.

[AIS1993] Agrawal, R. Imieliński, T. and Swami, A. "Mining association rules between sets of items in large databases". Proceedings of the 1993 ACM SIGMOD international conference on Management of data. p. 207. 1993.

[AL1994] William W. and Adams, Philippe Loustaunau. An Introduction to Gröbner Bases. American Mathematical Society, Graduate Studies in Mathematics, Vol. 3.1994.

[All1981] James F. Allen. An interval-based representation of temporal knowledge. In IJCAI'81: Proceedings of the 7th international joint conference on Artificial intelligence, pages 221–226, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[All1983] James F. Allen. Maintaining knowledge about temporal intervals. Communication of ACM, 26:832–843, 1983.

[All1984] James F. Allen. Towards a general theory of action and time.Artificial Intelligence, 23:123–154, 1984.

[Ami1977] Amir Pnueli. The temporal logic of programs. Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS), pp.46–57, 1977.

[Art1957] Arthur Prior . Time and Modality. Oxford University Press. 1957.

[Art1967] Arthur Prior. 'Logic, Modal'. In Edwards, P. (ed.) The Encyclopedia of Philosophy. New York: Macmillan. 1967.

[Art1968] Arthur Prior. Papers on Time and Tense. Oxford: Clarendon Press. 1958.

[APP1985] Apple Computer. Inc. Inside Macintosh. Addison-Wesley, 1985.

[Ben1983] Van Benthem, J., The Logic of Time, Kluwer Academic, Dordrech, 1983

[Ben1986] Ben Moszkowski. Executing Temporal Logic Programs. Cambridge University Press, Cambridge, England, 1986.

[BFG1990] H. Barringer, M. Fisher, D. M. Gabbay, G. Gough, and R. Owens. Metatem. A framework for programmingin temporal logic. In J. W. de Bakker, W. P. de Roever, and G. Rozenberg, editors, Stepwise Refinement of Distributed Systems, Models, Formalisms, Correctness, REX Workshop, Mook, The Netherlands, May 29 - June 2, 1989, Springer, Proceedings, v.430, pages 94–129., 1990.

[BK2005] Boguraev B and Ando RK. TimeML-Compliant Text Analysis for Temporal Reasoning.IJCAI.pages 997-100, 2005.

[BL2008] Pang B and Lee L. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2008.

[Bru1972] Bruce, B., A Model for Temporal References and Application in a Question Answering Program, Artificial Intelligence, 3, pp.1-25, 1972.

[BTK1991] Bacchus F., Tenenberg J. and Koomen J. A. A non-reified temporal logic, Artificial Intelligence, 52, pp.87-108,1991.

[Car1981] Carlson, L. Aspect and qualification, special issue on tense and aspect, in P. J. Tedeschi and A. Zaenen (eds.), Syntax and Semantics, 14,pp. 31-64,1981.

[Cha2014] Charles F. Manski. COMMUNICATING UNCERTAINTY IN OFFICIAL ECONOMIC STATISTICS. Department of Economics and Institute for Policy Research . Northwestern University. 2014

[Chi2001] Chittaro, L. Information Visualization and its Application to Medicine, Artificial Intelligence in Medicine Journal, 22(2): 81-88, 2001.

[Chr1992] Christian Freksa, Temporal Reasoning Based on Semi-Intervals, Artificial Intelligence 54. pp.199-227. 1992.

[CK1991] Cousins, S. B. and Kahn, G. The visual display of temporal information, Artificial Intelligence in Medicine, VOL(3), pp.341-357, 1991.

[CM2012] Angel X. Chang and Christopher D. Manning. SUTIME: A Library for Recognizing and Normalizing Time Expressions. 8th International Conference on Language Resources and Evaluation,2012.

[CMS1999] Card, S., MacKinlay, J. and Shneiderman, B. Readings in Information Visualization: Using Vision to Think. San Francisco: Morgan Kaufmann Publishers, 1999.

[CPH2008] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. "Introduction to Information Retrieval". Cambridge University Press. 2008.

[CPP1999] Combi, C., Portoni, L. and Pinciroli, F. Visualizing Temporal Clinical Data on the WWW, Proceedings Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM'99).Springer. pp. 301-311. 1999.

[CTR1992] Christian Freksa , Technische Universität München , Robert Fulton. Temporal Reasoning Based on Semi-Intervals. Artificial Intelligence.V.54,(1–2) , pp.199–227, 1992.

[DC2003] Dan Klein and Christopher D. Manning Fast Exact Inference with a Factored Model for Natural Language Parsing. In Advances in Neural Information Processing Systems 15, Cambridge, MA: MIT Press, pp. 3-10. 2003.

[DKC2003] Dan Klein and Christopher D. Manning. Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430. 2003.

[Dre1982] McDermott Drew, A Temporal Logic for Reasoning about Processes and Plans, Cognitive Science, 6, 101-155, 1982.

[DW1983] Dictionary of Computing. Oxford: Oxford University Press, 1983.

[EDE2012] Eugene Santos Jr. Deqing Lia.Eunice E. Santosb. John Korahb. Temporal Bayesian Knowledge Bases – Reasoning about uncertainty with temporal constraints. Expert Systems with Applications. V.39(17), pp.12905–12917. 2012.

[FB2000] Federico, Barber. Reasoning on Interval and Point-based Disjunctive Metric Constraints in Temporal Contexts. Journal of Artificial Intelligence Research. 2000.

[FDM2005] Jenny Finkel, Shipra Dingare, Christopher Manning, Malvina Nissim, Beatrice Alex, and Claire Grover. Exploring the Boundaries: Gene and Protein Identification in Biomedical Text. BMC Bioinformatics 6. 2005.

[FK1983] Winthrop Nelson Francis and Henry Kucera , Frequency Analysis of English Usage: Lexicon and Grammar. Houghton Mifflin. 1983.

[FT1993] Fudenberg, D. and Tirole, J. Game Theory, MIT Press, 1993.

[FWY2008] Yao TF,Cheng XW,Xu FY,Uszkoreit H Wang R. A survey of opinion mining for texts. Journal of Chinese Information Processing. 2008.

[Gab1989] D. M. Gabbay. The declarative past and imperative future: Executable temporal logic for interactive systems. In B. Banieqbal, H. Barringer, and A. Pnueli, editors, Temporal Logic in Specification, Altrincham, UK, April 8-10, 1987,Proceedings, volume 398 of Lecture Notes in Computer Science, pp.409–448. Springer, 1989.

[Gal1990] Galton, A., A Critical Examination of Allen's Theory of Action and Time, Artificial Intelligence, 42, 159-188. 1990.

[Gal1996] Galton A. P., An Investigation of Non-intermingling Principles in Temporal Logic, Journal of Logic and Computation, V.6(2), pp.267-290,1996.

[GB2003] Gábor Nagypál , Boris Motik. A fuzzy model for representing uncertain, subjective and vague temporal knowledge in ontologies. On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE Lecture Notes in Computer Science, V.2888, pp.906-923, 2003.

[GDH1998] Goldberg, David E.; Holland, John H.. "Genetic algorithms and machine learning". Machine Learning 3 (2), pp.95–99.1998.

[GS2010] Alfonso Gerevini, Alessandro Saetti. Computing the minimal relations in point-based qualitative temporal reasoning through metagraph closure. Artificial Intelligence.V.175(2),pp.556-585. February, 2011.

[Ham1971] Hamblin C. L. Instants and Intervals, Studium Generale 24, 127-134, 1971.

[HHD2003] Hua-Ping Zhang, Hong-Kui Yu, De-Yi Xiong, and Qun Liu. "HHMM-based Chinese lexical analyzer ICTCLAS." In Proceedings of the second SIGHAN workshop on Chinese language processing ,Association for Computational Linguistics,V(7), pp. 184-187. 2003.

[HL2002] Hua-ping, Zhang, and Liu Qun. "ICTCLAS." Insti-tute of Computing Technology, Chinese Academy of Sciences: 2002.

[HS1991] Halpern, J. and Shoham, Y. A Propositional Model Logic of Time Intervals, Journal of the Association for Computing Machinery, 38(4), 935-962.1991.

[IL1994] Tomasz Imielinski and Witold Lipski. Incomplete Information in

Relational Databases.Journal of the ACM, 31(4),pp. 761-791. 1994.

[JAS2002] M. D. Jaere, A. Aamodt and P. Skalle: ―Representing Temporal Knowledge for Case-Based Prediction‖. In Proceedings of the 6th European Conference on Advances in Case-Based Reasoning (ECCBR ‘02), Aberdeen, Scotland, UK, Sep 4-7, Vol. 2416, pp.174 - 188, 2002.

[JBM2010] Jixin Ma, Brian Knight, Miltos Petridis, Xiao Bai, "A Graphical Representation for Uncertain and Incomplete Temporal Knowledge," gcis, vol. 1, pp.117-120, 2010 Second WRI Global Congress on Intelligent Systems, 2010.

[JJ2008] Huang XJ,Zhao J. Sentiment analysis for Chinese text. Communications of CCF,(02) ,2008.

[KM1992] B. Knight, J. Ma: A General Temporal Model Supporting Duration Reasoning, Artificial Intelligence Communication, Vol.5(2), 75-84, 1992.

[KTI2002] Dan Klein, Kristina Toutanova, H. Tolga Ilhan, Sepandar D. Kamvar, and Christopher D. Manning. Combining Heterogeneous Classifiers for Word-Sense Disambiguation. In Workshop on Word Sense Disambiguation: Recent Successes and Future Directions at ACL 40, pp. 74-80.2002.

[KH1995] Hee-Cheol, Kim . Prediction and Postdiction under Uncertainty , Baltzer Journals,2 July,2002.

[KM2001] Kosara, R. and Miksch, S. Metaphors of Movement: A Visualization and User Interface for Time-Oriented, Skeletal Plans, to appear in Artificial Intelligence in Medicine, Special Issue: Information Visualization in Medicine, 2001.

[KS1986] Kowalski, R. and Sergot, M., A logic-based calculus of events, New Generation Computing, 4, pp.67-95 ,1986.

[KTI2002] Dan Klein, Kristina Toutanova, H. Tolga Ilhan, Sepandar D. Kamvar, and Christopher D. Manning. Combining Heterogeneous Classifiers for Word-Sense Disambiguation. In Workshop on Word Sense Disambiguation: Recent Successes and Future Directions at ACL 40, pp. 74-80. 2002.

[KWL2009] Kalvala, Sara, Warburton, Richard and Lacey, David. Program transformations using temporal logic side conditions. ACM Transactions on Programming Languages and Systems, V. 31 (4). 2009.

[Lad1987] Ladkin, P., Models of axioms for time intervals, In Proceedings of the 6th National Conference on Artificial Intelligence. pp. 234-239.1987.

[Lad1992] Ladkin, P., Effective solutions of qualitative intervals constraint problems, Artificial Intelligence, 52, 105-124.1992.

[LB2002] Edward Loper, Steven Bird. NLTK: The Natural Language Toolkit. Computation and Language. 2002.

[Li1997] Minglu Li. Temporal Models for Multimedia Synchronization. Computer Science and Engineering Shanghai Jiao Tong University Shanghai. 1997.

[Lie2009] Lieberman, Henry. User Interface Goals, Al Opportunities, AI Magazine, V. 30(4), pp.16-22, 2009.

[MA1988] Joel McCormack and Paul Asente. ‘‘An Overview of the X Toolkit,’’ Proceedings of the ACM SIGGRAPH Symposium on User Interface Software,’’ Banff, Alberta, Canada.10(17-19). pp.46-55, 1988.

[Ma2007] J. Ma: Ontological Considerations of Time, Meta-Predicates and Temporal Propositions, Applied Ontology, Vol.2(1), pp.37-66, 2007.

[MAN2011] Christopher D. Manning. 2011. Part-of-Speech Tagging from 97% to 100%: Is It Time for Some Linguistics? In Computational Linguistics and Intelligent Text Processing, 12th International Conference, CICLing, Proceedings, Part I. 2011.

[MBC2006] Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. 2006.

[MBM2008] J. Ma, B. Knight, M. Petridis: Characterizing the Most General Temporal Constraints, Proc. ECAI-08 Workshop on Spatial and temporal Reasoning, pp.26-30, 2008.

[MC2000] Christopher D. Manning and Bob Carpenter. Probabilistic Parsing Using Left Corner Language Models. In Harry Bunt and Anton Nijholt (eds), Advances in Probabilistic and Other Parsing Technologies. Kluwer Academic Publishers, pp. 105-124. 2000.

[Mer1995] S. Merz. Efficiently executable temporal logic programs. In M. Fisher and R. Owens, editors, Executable Modal and Temporal Logics, IJCAI ’93, Workshop, Chamb´ery, France, August 28, 1993, Proceedings, volume 897 of Lecture Notes in Computer Science,pp. 69–85. Springer, 1995.

[MH2006] Ma J and Hayes P. “Primitive Intervals Vs Point-Based Intervals: Rivals Or Allies?”, the Computer Journal, 49(1), pp.32-41, 2006.

[MK1994] Ma, J. and Knight, B., A General Temporal Theory, the Computer Journal, 37(2), 114-123.1994.

[MK1996] Ma, J. and Knight, B., A Reified Temporal Logic, the Computer

Journal, 39(9), 800-807,1996.

[MK2003] Ma, J. and Knight, B., Representing The Dividing Instant, the Computer Journal, 46(2), pp.213-222,2003.

[MKP1994] Ma J., Knight B. and Petridis M. A Revised Theory of Action and Time based on Intervals and Point, The Computer Journal, V.37(10), pp.847-857, 1994.

[MLM2010] Jianbing Ma, Weiru Liu and Paul Miller. Event modelling and reasoning with uncertain information for distributed sensor networks, In Proceedings of the 4th international conference on Scalable uncertainty management, pp.236-249,2010.

[MSM1993] Marcus,M.,Santorini,B.,Ann Marcinkiewicz,A. Building a Large Annotated Corpus of English: the Penn Treebank. Computational Linguistics 19: pp. 313-330,1993.

[MST994] Michie, D.Spiegelhalter, D. J.Taylor. Machine Learning, Neural and Statistical Classification. Ellis Horwood. 1994.

[PHD2009] Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. Discriminative Reordering with Chinese Grammatical Relations Features. In Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation.2009.

[PMR1996] Plaisant, C., Milash, B. Rose, A., Widoff, S. and Shneiderman, B. LifeLines: Visualizing Personal Histories, Proceedings of the CHI ’96 Conference on Human Factors in Computing Systems, ACM Press, pp. 221-227, 1996.

[RAJ2011] Rajaraman, A.; Ullman, J. D "Data Mining". Mining of Massive Datasets. pp. 1–17. 2011.

[RC2003] Roger Levy and Christopher D. Manning. Is it harder to parse Chinese, or the Chinese Treebank? .ACL 2003, pp. 439-446. 2003.

[RH1998] Mateescu, R., Garavel, H.: XTL: A Meta-Language and Tool for Temporal Logic Model-Checking. In: Proceedings of the International Workshop on Software Tools for Technology Transfer STTT 1998, Aalborg, Denmark. 1998.

[RJC2013] Richard Socher, John Bauer, Christopher D. Manning and Andrew Y. Ng. Parsing With Compositional Vector Grammars. Proceedings of ACL. 2013.

[RLM2008] Rokach. Lior. Maimon. Data mining with decision trees: theory and applications. World Scientific. 2008.

[SAC2011] Valentin I. Spitkovsky, Hiyan Alshawi, Angel X. Chang, and Daniel Jurafsky. 2011. Unsupervised Dependency Parsing without Gold Part-of-Speech Tags. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. 2011.

[SAJ1012] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2012. Bootstrapping Dependency Grammar Inducers from Incomplete Sentence Fragments via Austere Models. In Proceedings of the 11th International Conference on Grammatical Inference. 2012.

[SAJ2013] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. Breaking Out of Local Optima with Count Transforms and Model Recombination: A Study in Grammar Induction. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013.

[SC2000] Shahar, Y. and Cheng C. Model-Based Visualization of Temporal Abstractions. Computational Intelligence; 16(2): pp.279-306, 2000.

[SC2010] Spence Green and Christopher D. Manning. 2010. Better Arabic Parsing: Baselines, Evaluations, and Analysis. 2010.

[SE1991] Shapiro, Ehud Y. Inductive inference of theories from facts, Research Report 192, Yale University, Department of Computer Science, 1981. Reprinted in J.-L. Lassez, G. Plotkin (Eds.), Computational Logic, The MIT Press. pp.199 – 254. 1991.

[Sho1987] Shoham Y. Reified Temporal Logics: Semantical and Ontological Considerations, Advances in Artificial Intelligence - II, B. Du Boulay, D. Hogg and L. Steels (Editors), Elsevier Science Publishers B. V. (North-Holland), pp.183-190, 1987.

[Sho1987] Y. ShohamTemporal Logics in AI: Semantical and Ontological Considerations, Artificial Intelligence, 33, 89-104, 1987.

[SIB2010] Mohamed Soliman, Ihab Ilyas and Shalev Ben-David. Supporting ranking queries on uncertain and incomplete data. The International Journal on Very Large Data Bases Archive. V.19(4), pp. 477-501. 2010.

[SKL2014] Richard Socher.Andrej Karpathy.Quoc,V. Le.Christopher D. Manning, and Andrew Y. Ng. Grounded Compositional Semantics for Finding and Describing Images with Sentences. Transactions of the Association for Computational Linguistics. 2014.

[SMJC2011] Spence Green, Marie-Catherine de Marneffe, John Bauer, and Christopher D. Manning. 2010. Multiword Expression Identification with Tree Substitution Grammars: A Parsing tour de force with French.. 2011.

[Sut1963] Sutherland.IvanE. ‘‘SketchPad: A Man-Machine Graphical Communication Sysem,’’ AFIPS Spring Joint Computer Conference.

23. pp. 329-346. 1963.

[Tod1993] Todd Mansell. A method for planning given uncertain and incomplete information, In Proceedings of the Ninth international conference on Uncertainty in artificial intelligence, pp. 350-358, 1993.

[Vil1994] Vila, L. A survey on temporal Reasoning in Artificial Intelligence, AI Communication, 7:4-28, 1994.

[VJ2012] Rob Voigt and Dan Jurafsky. Towards a Literary Machine Translation: The Role of Referential Cohesion. In Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature. 2012.

[WM2010] Mengqiu Wang and Christopher D. Manning. Probabilistic Tree-Edit Models with Structured Latent Variables for Textual Entailment and Question Answering. In Proceedings of the 23rd International Conference on Computational Linguistics. 2010.

[WMK2013] Y. Wang, J. Ma and B. Knight: Implementing a Consistency Checker for Uncertain or Incomplete Temporal System, International Journal of Hybrid Information Technology, 6(6), pp.195-202. 2013.

[WMK2014] Yue, Wang, and Ma Jixin. "A Computer Aided Case Tool for Temporal Knowledge Visualization." 3rd International Conference on Computer Science and Service System. Atlantis Press. 2014.

[XG2004] Xing, Wenpu. Ali Ghorbani. "Weighted PageRank Algorithm." Communication Networks & Services Research .proceedings.second Annual Conference on. IEEE Computer Society, pp.305-314. 2004.

[Yos1999] William A. Yost. Temporal models of pitch processing. The Journal of the Acoustical Society of America. 105(2). 1999.

[ZD2003] Zhou Lina, and Dongsong Zhang. "NLPIR: A theoretical framework for applying natural language processing to information retrieval." Journal of the American Society for Information Science and Technology 54.2: 115-123, 2003.

[ZSC2013] Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. Bilingual Word Embeddings for Phrase-Based Machine Translation. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013.

APPENDIX A

Programming code

1. Social JS code:

```
itisnavi = function(context){


    itisnavi.prototype.context = context;

    this.pagiFlag = false;

    this.acunumber = 0;

    this.users = new Array();

    itisnavi.prototype.users = this.users;

    this.initialize();


}


itisnavi.prototype = {


      initialize :    function(){

                var   str='<input   id="naviinput1"   type="button"   value=" 搜 索 地 点   Input
    location"/>'+

                            '<input id="naviinput2" type="button" value="查看街景 Streetview" />'+


                            '<input id="naviinput3" type="button" value="开始找人 Search" />'+

                            '<input id="naviinput4" type="button" value="淘客 Shop finder" />'+

                            '<input id="naviinput5" type="button" value="帮助 help "    />';
```

```
        document.getElementById("navi").innerHTML = str;
        itisnavi.prototype.ShowLoginView();

    },
    eventbind :function(){
        $("#naviinput1").bind("click",itisnavi.prototype.searchPlace);
        $("#naviinput2").bind("click",itisnavi.prototype.streetView);
        $("#naviinput3").bind("click",itisnavi.prototype.accurateusrs);
        $("#naviinput4").bind("click",itisnavi.prototype.showProfesView);
        $("#naviinput5").bind("click",itisnavi.prototype.showhelp);
    },
    showProfesView :function(){
        new Profession().showProfesView();
    },
    showhelp: function(){
        if(!$("#helpdialog")[0])
        {
            $("body").append(
                '<div id="helpdialog" style="text-align:left">'+
                 '<p>'+
                 '<h1><1>操作区 draw cycle</h1>'+
                        '地图上的圆为当前操作区域,圆可放大缩小,最大半径10000米。可用鼠标拖动圆，或是移动地图改变当前选区。User can draw and move cycle on Google map, the largest Radius of cycle is 10000 meters'+
```

```
'</p>'+
 '<P>'+
   '<h1><2>搜索功能 search</h1>'+
     '在搜索界面输入目的地，点击“搜索”，地图会自动切换到当前区域。click search,map will go to the location you choose'+
 '</p>'+
  '<p>'+
   '<h1><3>查看街景 streetview</h1>'+
     '显示距离操作区圆心最近位置的街景，需要查看某地街景就把操作区的圆心移动到相应位置。help user to check the street view of chosen location'+
  '</p>'+
  '<p>'+
   '<h1><4>查找用户 user finder</h1>'+
     '显示过去 12 小时，在该区域发过微博的用户,load the users who has sent tweets over past 12 hours'+
  '<p>'+
  '<input type="button" value="知道了 OK" id="cancelbut"><input type="button" value="注销 cancel" id="logout">'+
'</div>'
);
$("#cancelbut").bind("click",function(){$("#helpdialog").dialog("close")});
$("#logout").bind("click",itisnavi.prototype.logout);
var width = $("body").width();
  var left = 0;
  var height = $("body").height();
  var top =  0
```

```
            itisnavi.prototype.darwDialog($("#helpdialog"),top,left,width,height);
        }



},
logout: function(){
        $.ajax({
            url :'logout',   //后台处理程序
            type:'post',      //数据发送方式
            dataType:'json',    //接受数据格式
            success: function(oResponse){
                WB2.logout(function(){
                    window.open('http://www.yifenr.com/mmlogin.jsp','_self');
                });
            }

        });
},
streetView : function(){

    var viewservice =   new google.maps.StreetViewService();
    var center = itisnavi.prototype.context.circle.getCenter();
    var radius = parseInt(itisnavi.prototype.context.circle.getRadius());

    viewservice.getPanoramaByLocation(center,radius,itisnavi.prototype.streetViewCbk);
```

```
},

streetViewCbk: function(StreetViewPanoramaData,StreetViewStatus){


    switch (StreetViewStatus){

        case google.maps.StreetViewStatus.OK:

            var viewid    = StreetViewPanoramaData.location.pano;

            var panoramaOptions = {

                    pano: viewid,

                     pov: {

                        heading: 34,

                        pitch: 10

                     }

                  };


            if(!$("#pano")[0])

            {

                $("body").append(

                    '<div id="pano">'+


                    '</div>'

                );

                  $("#hello").before(

                        '<div                                         id="topCoverDiv"
style="width:auto;height:auto;top:100px;position:absolute;z-index:1000;border:1px
solid:#0000ff;color:#00ff00;">'+

                           '<a              id="backtomap"              href="javascript:;"
```

```
style="color:#00ff00">返回</a>'+
                    '</div>');
                                    var left = $("body").width()*0.9;
                  left = left +'px';
                  $("#topCoverDiv").css("left", left);
                                    $("#topCoverDiv").css("top", "10px");

                  var height = $("body").height();
                  itisnavi.prototype.darwDialog($("#pano"),0,0,"100%",height);



$("#backtomap").bind("click",function(){$("#pano").dialog("close");$("#topCoverDiv").remo
ve()});
                  var panorama = new
google.maps.StreetViewPanorama(document.getElementById('pano'),panoramaOptions);
                  itisnavi.prototype.context.map.setStreetView(panorama);
              }
              break;
          case google.maps.StreetViewStatus.UNKNOWN_ERROR:
              alert("未知错误,请刷新页面");
              break;
          case google.maps.StreetViewStatus.ZERO_RESULTS:
                  alert("区域内无街景");
              break;
          default:
              break;
```

```
        }


    },
    startSearch : function(){
            if($("#target").val()==""){
                alert("地点输入不能为空 location cannot be none");
            }
            else{
                $("#searchresult").html("<img src=img/ref.gif>");
                var request = {
                        address:$("#target").val()
                }
                var service =new google.maps.Geocoder();
                service.geocode(request, itisnavi.prototype.callback);
            }


    },


    callback : function(results, status){
            if (status == google.maps.GeocoderStatus.OK) {
                $("#searchresult").html("");
                    for (var i = 0; i < results.length; i++) {
                        $("#searchresult").append('<a id="'+i+'"
style="dispaly:line-block;margin-top:20pxpx;border-bottom:1px solid #aaaaaa;"
href="javascript:;")>'+results[i].formatted_address+'</a><br>');
```

```
$("#"+i).bind("click",itisnavi.prototype.showStreetView(results[i].geometry.location,result
s[i].formatted_address));


                }
                if(i==1)
                {
                    $("#0").trigger("click");
                }
            //    itisnavi.prototype.context.map.setCenter(results[0].geometry.location);
            }
        else if(status == google.maps.GeocoderStatus.ERROR){
            alert("There was a problem contacting the Google servers try again");
            $("#searchresult").html("");
        }
        else if(status == google.maps.GeocoderStatus.INVALID_REQUEST){
            alert("This GeocoderRequest was invalid.");
            $("#searchresult").html("");
        }
        else if(status == google.maps.GeocoderStatus.OVER_QUERY_LIMIT){
            alert("The webpage has gone over the requests limit in too short a period of
time.");
            $("#searchresult").html("");
        }
        else if(status == google.maps.GeocoderStatus.REQUEST_DENIED){
            alert("The webpage is not allowed to use the geocoder.");
            $("#searchresult").html("");
```

```
            }
            else if(status == google.maps.GeocoderStatus.UNKNOWN_ERROR){
                alert("A geocoding request could not be processed due to a server error. The request may succeed if you try again.");
                $("#searchresult").html("");
            }
            else if(status == google.maps.GeocoderStatus.ZERO_RESULTS){
                alert("No result was found for this GeocoderRequest.");
                $("#searchresult").html("");
            }
    },
    showStreetView : function( location,address)
    {
        return function(){
                itisnavi.prototype.context.map.setCenter(location);
                itisnavi.prototype.context.circle.setCenter(location);
                var infowindow = new google.maps.InfoWindow();
                 var marker = new google.maps.Marker({
                        map: itisnavi.prototype.context.map,
                        position: location,
                        title: address
                    });

                $("#searchdialog").dialog("close")
```

```
        }
    },
     createMarker : function(place) {
            var infowindow = new google.maps.InfoWindow();
            var marker = new google.maps.Marker({
               map: itisnavi.prototype.context.map,
               position: place.geometry.location,
               title:place.formatted_address
            });
        },
    searchPlace :function(){
        if(!$("#searchdialog")[0])
        {
             $("body").append(
                  '<div id="searchdialog" style="text-align:center;height:30px">'+
                       '<input id="target" type="text" placeholder="输入要查找的地址 input the location you want to search" style="width:96%;">'+
                       '<input type="button" value="搜索 locate" id="searchbut"/><input type="button" value="退出 cancel" id="cancelbut">'+
                       '<div id="searchresult" style="width:100%;height:auto;"></div>'+
                  '</div>'

             );
             $("#cancelbut").bind("click",function(){$("#searchdialog").dialog("close")});
             $("#searchbut").bind("click",itisnavi.prototype.startSearch);
```

```
$("#searchdialog").bind("keydown",function(event){if(event.keyCode
==13){$("#searchbut").trigger("click")}});

var defaultBounds = new google.maps.LatLngBounds(

        new google.maps.LatLng(-90, 151.1759),

        new google.maps.LatLng(90, 151.2631));

var input = (document.getElementById('target'));

var searchBox = new google.maps.places.SearchBox(input);


var map = itisnavi.prototype.context.map;

  var markers = [];

  google.maps.event.addListener(searchBox, 'places_changed', function() {

        var places = searchBox.getPlaces();

        var bounds = new google.maps.LatLngBounds();




          var marker = new google.maps.Marker({

            map: map,

            title: places[0].formatted_address,

            position: places[0].geometry.location

          });

        itisnavi.prototype.context.map.setCenter(places[0].geometry.location);

        itisnavi.prototype.context.circle.setCenter(places[0].geometry.location);

        $("#searchdialog").dialog("close");

      });
```

```
            var    widthpercent = 0.8;
            var width = $("body").width()*widthpercent;
                var left = $("body").width()*(1-widthpercent)*0.5;



                var heightpercent = 0.3;
                var height = $("body").height()*heightpercent;
                var top =    $("body").height()*0.05;

               itisnavi.prototype.darwDialog($("#searchdialog"),top,left,width,"auto");
      }

},
accurateusrs : function(name,reminTImes){
      var pagination = new Pagination(1,"acu");
      var pagiHtml =    pagination.buildHtml();

     if(!$("#dialog-users")[0])
     {
            var lat = itisnavi.prototype.context.circle.getCenter().lat();
            var lng = itisnavi.prototype.context.circle.getCenter().lng();
            var radius = itisnavi.prototype.context.circle.getRadius();
```

```
$("body").append(
    '<div id="dialog-users" style="border:1px solid #0000ff;z-index:100; background:#fff;height:100%" >'+
        '<div style="text-align:left;width:100%;border:1px solid #00ff00" >所选区域 圆心 coordinates：(<span id="lat" class="red">'+lat+',</span><span id="lng" class="red">'+lng+'</span>) 半径 Radius：<span id="radius" class="red">'+radius+'</span>米 区域内用户 number of choosen users：<span id="areacount" class="blue" ></span></div>'+
        '<div style="text-align:left"><span style="text-align:left;font-size:13px;font-weight:bold">选择推广内容 choosen tweets you want to choose for sending： <input id="selecttimeline" type="button" value="从我的微博选择 from my tweets">  <input id="selectTimelinebyUrl" type="button" value="任意微博 any tweets" \></span>'+
            '<p id="share_text"  style="width:100%;height:auto"></p>'+
            '<img id="thump" style="margin:2px;height:auto;width:auto"/><br>'+
            '<input id="comment" type="text" placeholder="输入评论不超过 100 字 less than 100 words" style="width:30%">'+
        '</div>'+
        '<div style="text-align:left; font-size:13px;font-weight:bold;background:#fff;">性别 gender '+
            '<select name="sex1" id="sex1" class="text ui-widget-content ui-corner-all">' +
                '<option value="0">不限 none</option>'+
                '<option value="1">女 female</option>'+
                '<option value="2">男 male</option> '+
            '</select>'+
            '  年龄段 ages:'+
                '<input id="agestart" type="text" placeholder="1" style="width:30px" \>-'+
              '<input id="ageend" type="text" placeholder="100"
```

```
style="width:30px" \>'+
                    '排序方式 sort by：<select name="sort" id="sort" class="text ui-widget-content ui-corner-all">' +
                        '<option value="0">时间排序 time</option>'+
                        '<option value="1">地点排序 location</option>'+
                    '</select>'+
                    '  <a id="filter" href="javascript:;" style="cursor:pointer" ><font color="#00aa00">筛选 filter</font> </a>'+
                    '  点击头像选中感兴趣的用户/点击昵称查看主页 click the profile picture to choose：'+
                    '<input id="currentpageall" type="checkbox" value="当前页全选"\><label>当前页全选 choose all</label>'+
                '</div>'+
                '<div class="friendtabdiv" style="border:1px solid #0000ff;height:auto;overflow:auto">'+



                '</div>'+
                pagiHtml+
                '<div style="text-align:left;font-size:13px;font-weight:bold"> 今日可推 number of tweets you can send：<span id="totalcount">'+new Util().getCookie('remainTimes') +'</span>人 当前选中:<span id="selectcount">'+itisnavi.prototype.users.length+'</span>人 还可选择 numbers left ：<span id="reservecount">'+(parseInt(new Util().getCookie('remainTimes'))-itisnavi.prototype.users.length)+'</span>人'+
                    '  <a id="clearall" href="javascript:;" style="cursor:pointer" ><font color="#00aa00">清空 clear</font> </a>'+
                '</div>'+
                '<div id="friendstotal" class="ui-widget"></div>'+
                '<div><input id="selectokbut" type="button" value="开始推送
```

```
start"></input><input id="reback" type="button" value="返回 cancel"></input></div>'+
        '</div>'


        );
        $("#selecttimeline").button();
        $("#selectTimelinebyUrl").button();


        $("#selectokbut").bind("click",itisnavi.prototype.areasendhandle);
        $("#reback").bind("click",function(){ $( "#dialog-users").dialog("close");});
        $("#selecttimeline").bind("click",itisnavi.prototype.selecttimeline);
        $("#filter").bind("click",itisnavi.prototype.filter);
        $("#currentpageall").bind("click",itisnavi.prototype.currentpageall);
        $("#selectTimelinebyUrl").bind("click",itisnavi.prototype.selectTimelinebyUrl);
        $("#clearall").bind("click",function(){
            var len = $("#friendstotal").children("p").length;

            for(var i=0; i<len; i++){
                $("#friendstotal").children("p").eq(0).children("img").trigger("click");
            }
        });
        itisnavi.prototype.showArray();



        var height = $("body").height();
        itisnavi.prototype.darwDialog($("#dialog-users"),0,0,"100%",height);
```

```
            pagination.buildLsn(itisnavi.prototype,itisnavi.prototype.showaccuratedetail);
            itisnavi.prototype.pagiFlag = true;
            itisnavi.prototype.showaccuratedetail(1);
            //$("#map-canvas").css("warp","none");

  }
},
currentpageall : function(e){
  var checked    = $(e.currentTarget).is(':checked');
  var len = $(".friendtabdiv").children("div").length;
  for(var i=0; i<len; i++){

            var rgb = $(".friendtabdiv").children("div").eq(i).css('background-color');
            if(rgb.substr(0,3) =='rgb')
            {
                rgb = rgb.match(/^rgb\((\d+),\s*(\d+),\s*(\d+)\)$/);
                function hex(x) {return ("0" + parseInt(x).toString(16)).slice(-2);}
                rgb= "#" + hex(rgb[1]) + hex(rgb[2]) + hex(rgb[3]);
            }
            if(checked){
                if(rgb == "#f9f9f9"){
                    $(".friendtabdiv").children("div").eq(i).children("img").trigger("click");
```

```
                    if(parseInt($("#selectcount").html()) >=
parseInt( $("#totalcount").html())){
                        break;
                    }

                }
            }
            else{
                if(rgb == "#548abe"){
                    $(".friendtabdiv").children("div").eq(i).children("img").trigger("click");

                }
            }
        }
    },
    selecttimeline : function(){
        if(!$("#dialog-timeline")[0])
        {
            $("body").append(
                    '<div id="dialog-timeline" class="dialog" title="选择我的微博 choose
from my tweets">'+
                    '<p style="font-weight:bold;font-size:10px;width:100%">从最近的 40 条
微博选择一条自己的微博 latest 40 tweets <input style="float:right" id="timelinereback"
type="button" value="返回" \></p><br/>'
                );

$("#timelinereback").bind("click",function(){$( "#dialog-timeline").dialog("close");});
```

```
WB2.anyWhere(function(W){
        // 获取微博


W.parseCMD("/statuses/user_timeline.json", function(sResult, bStatus){
        if(bStatus == true) {
                        // alert("success");
                          Result= sResult;
                    // alert(sResult.total_number);
                      var number = sResult.total_number<40?sResult.total_number:40;
                    // alert("number:"+number);
                      var html;
                      for(var i=0;i<number;i++)
                      {
                             if(sResult.statuses[i].thumbnail_pic==undefined)
                             {
                                    html = '<div id="'+sResult.statuses[i].id+'" style="border-bottom:1px solid #C8C8C8;text-align:left;font-size:10px">'+
                                                   '<p style="width:100%">'+sResult.statuses[i].text+'</p>'+
                                                   '<span style="width:100%;text-align:right" >'+sResult.statuses[i].created_at+'</span>'+
                                                 '<div style="text-align:right"><input type="button" value="选择 ok" class="choosebut" style="background-color:#4bacff;width:50px;height:25px; float:right;text-align:center;margin:3px"></div>'+
                                                   '</div>';
                             }
                             else
```

```
                    {
                        html = '<div id="'+sResult.statuses[i].id+'" style="border-bottom:1px solid #C8C8C8;text-align:left;font-size:10px">'+
                            '<p style="width:100%">'+sResult.statuses[i].text+'</p>'+
                            '<img src="'+sResult.statuses[i].bmiddle_pic+'" style="margin:2px;height:auto;width:auto"/>'+
                            '<span style="width:100%;text-align:right">'+sResult.statuses[i].created_at+'</span>'+
                            '<div><input type="button" value="选择" class="choosebut" style="background-color:#4bacff;width:50px;height:25px; float:right;text-align:center;margin:3px"></div>'+
                            '</div>';
                    }
                    $("#dialog-timeline").append(html);
                }
                $(".choosebut").button();
                $(".choosebut").bind("click",itisnavi.prototype.timelineselect);
            }
        },{
            uid :user.id,
            source:appkey,
            count:40,
            feature:1
        },{
            method: 'get'
```

```
        });
        });

        var height = $("body").height();
        itisnavi.prototype.darwDialog($("#dialog-timeline"),0,0,"100%",height);
       }

    },
    selectTimelinebyUrl: function(e){
        if(!$("#selectTimelinebyUrlDialog")[0]){
            var html = '<div id="selectTimelinebyUrlDialog">'+
                '<input id="timelineurl" placeholder="微博地址 microblog address" style="width:90%;float:left" type="text" \><input id="relativecount" placeholder="关联账号 relate account" type="text" style="float:left;width:90%" \>'+
                '<input id="ok" type="button" value="确定 ok"><input id="cancel" type="button" value="取消 cancel">'
                '</div>'
        $("body").append(html);
$("#cancel").bind("click",function(){$("#selectTimelinebyUrlDialog").dialog("close");});
            $("#ok").bind("click",itisnavi.prototype.timelinebyUrlDialogOK);
            var obj = $(e.currentTarget);
            var left =obj.offset().left;
            var top = obj.offset().top;
```

```
            itisnavi.prototype.darwDialog($("#selectTimelinebyUrlDialog"),top,left,500,150);


        }
    },
    timelinebyUrlDialogOK : function(e){
        var timelineId = new Util().sinaWburl2ID($("#timelineurl").val());
        WB2.anyWhere(function(W){
            // 获取微博


        W.parseCMD("/statuses/show.json", function(sResult, bStatus){
            if(bStatus == true) {
                $("#share_text").html(sResult.text);
                $("#share_text").attr("name",sResult.id);
                $("#thump").attr("src",sResult.bmiddle_pic);


            }else{
                alert("输入微博地址有误，请输入正确微博地址");
            }


        },{


            source:appkey,
              id:timelineId
        },{
            method: 'get'
```

```
    });
    });
    $("#selectTimelinebyUrlDialog").dialog("close");

},
timelineselect : function(e){



      var obj = $(e.currentTarget);
      $("#share_text").html( obj.parent().parent().children("p").html());
      $("#share_text").attr("name",obj.parent().parent().attr("id"));
      if(obj.parent().parent().children("img")[0]){
          var src = obj.parent().parent().children("img").attr("src");
          /*
          var html = '<img src="'+src+'" style="margin:2px;height:auto;width:auto"/>';
         $("#share_text").after(html);
         */
          $("#thump").attr("src",src);
      }
      else{

          $("#thump").removeAttr("src").attr("src",'');
      }
      $( "#dialog-timeline").dialog("close");
```

```
    },
    filter : function(){
        itisnavi.prototype.showaccuratedetail(1);
    },
    taskRunView : function(){
        var expire = parseInt(itisnavi.prototype.users.length/3);
        var html='';
        if(expire > 10){
            html = '任务正在执行中...预计耗时'+expire+'分钟,建议后台运行';
        }else{
            html = '任务正在执行中...预计耗时'+expire+'分钟';
        }
        var innerhtml = '<div id="waitDialog">'+
                                    '<img src="/img/ref.gif"\>'+
                                    '<p>'+html+'</p>'+
                                    '<input id="noDisplay" type="button"    value="后台运行">'+
                            '</div>';
        $("body").append(innerhtml);
      itisnavi.prototype.darwDialog($("#waitDialog"),200,300,"50%","auto");
      $("#noDisplay").button();
      $("#noDisplay").bind("click",function(){$("#clearall").trigger("click");
$("#dialog-users").dialog("close");$("#waitDialog").dialog("close");});
    },
```

```
taskEndView : function(sucNuber){
    var innerhtml = '<p>'+
                '<font color="#00ff00" size="14px"> 推 送 成 功 ： '+sucNuber+'</font>'+
                '</p>'+
                '<input id="close" type="button" value="关闭">';
    $("#waitDialog").html(innerhtml);
    $("#close").button();
$("#close").bind("click",function(){$("#waitDialog").dialog("close");$("#clearall").trigger("click"); $("#dialog-users").dialog("close")});
},
startRepostTask :function(){
      $("#ok").attr("disabled",true);
      var usrname = $("#usrname").val();
      var pwd = $("#pwd").val();
      var customer = {
                userName : usrname,
                passWord : pwd
            };
      var data = 'postData ='+JSON.stringify(customer)    ;
      $.ajax({
            url :'judgeAndRemain',
            type:'post',
            dataType:'json',
            data:data,
```

```
success:    function(oReponse){


            var reminTImes = oReponse.remainTimes;
            var spreading = oReponse.spreading;
            if(reminTImes == -1){
                    alert("密码不正确,");
            }else if(spreading ==1){
                     alert("您上次任务还没有推送结束，请稍后再试！");
            }else {
                    var timelineId = $("#share_text").attr("name");

                    if(typeof(timelineId) == "undefined"){
                            alert("请先选择一条微博");
                            return;
                      }
                    var arraydata = {
                            array: itisnavi.prototype.users,
                            id:    timelineId,
                            usrName:usrname,
                            comment:$("#comment").val()
                             };
                    var data = 'postData ='+JSON.stringify(arraydata);
                    $("#pwddialog").dialog("close");
                    itisnavi.prototype.taskRunView();
                    $.ajax({
```

```
url :'Json_Circle_startRepost',
type:'post',
dataType:'json',
data:data,
success:    function(oReponse){
itisnavi.prototype.taskEndView(oReponse.sucNuber);
    //alert("推送成功："+oReponse.sucNuber);

    var total = parseInt(new Util().getCookie('remainTimes'));
    total = total - parseInt(oReponse.sucNuber);
    new Util().setCookie('remainTimes',total);
    $("#totalcount").html(total);
$("#reservecount").html(parseInt( $("#totalcount").html())-parseInt($("#selectcount").html()));
        }
    });




            }
        }
    });
},
```

```
checkNamePwd :function(){
  var usrname = $("#usrname").val();
  var pwd = $("#pwd").val();


  var customer = {
                    userName : usrname,
                    passWord : pwd
        };
    var data = 'postData ='+JSON.stringify(customer)    ;
    $.ajax({
            url :'judgeAndRemain',
            type:'post',
            dataType:'json',
                data:data,
            success:   function(oReponse){


                        var reminTImes = oReponse.remainTimes;
                        if(reminTImes == -1){
                                alert("密码不正确,");
                        }else {
                        new Util().setCookie('name', usrname);
                        new Util().setCookie('pwd',pwd);
                        new Util().setCookie('remainTimes',oReponse.remainTimes);


                          $("#pwddialog").dialog("close");
```

```
                                        }
                                    }
                    });
},
  ShowLoginView : function()
  {
    if(!$("#pwddialog")[0]){
        var html ='<div id="pwddialog">'+
                          '<input id="usrname" style="width:100%;float:left" type="text"
placeholder="输入推送账号" \>'+
                            '<input id="pwd" style="width:100%;float:left" type="password"
placeholder="输入推送密码" \>'+
                            '<div><input id="ok" type="button" value="确认" \></div>'+
                    '</div>';

        $("body").append(html);
        $("#cancel").bind("click",function(){$("#pwddialog").dialog("close");});
        $("#ok").bind("click",itisnavi.prototype.checkNamePwd);
        itisnavi.prototype.darwDialog($("#pwddialog"),300,500,200,"auto");
        $("#pwddialog").bind("keydown",function(event){if(event.keyCode
==13){$("#ok").trigger("click")}});
    }
  },
  areasendhandle : function(){
    if(!$("#pwddialog")[0]){
```

```
var html ='<div id="pwddialog">'+
        '<input id="usrname" style="width:100%;float:left" type="text" placeholder="输入推送账号" \>'+
        '<input id="pwd" style="width:100%;float:left" type="password" placeholder="输入推送密码" \>'+
        '<div><input id="ok" type="button" value="确认" \><input id="cancel" type="button" value="取消" \></div>'+
    '</div>';


$("body").append(html);
$("#cancel").bind("click",function(){$("#pwddialog").dialog("close");});
$("#ok").bind("click",itisnavi.prototype.startRepostTask);
itisnavi.prototype.darwDialog($("#pwddialog"),300,500,200,"auto");
$("#pwddialog").bind("keydown",function(event){if(event.keyCode ==13){$("#ok").trigger("click")}});
}


/*
var names=''
    $("#friendstotal").children("p").each(function(i,n){
        names = names +'@'+ $(n).children("span").html();
    })
    names = names +'@'+$("#relativecount").val();
    var repoststatus = names;
    var record = $("#share_text").val();
    names = $("#share_text").val()+'___||我通过 http://www.yulanxy.com/mmlogin.jsp 在地图上找到你'+names;
```

```
$("#share_text").val(names);

if(typeof($("#thump").attr("src")) =='undefined'){
        WB2.anyWhere(function(W){
            W.parseCMD("/statuses/update.json", function(sResult, bStatus){
                if(bStatus == true) {
                    alert("发送成功");
                    $("#dialog-users").dialog( "close" );

                }
                else{
                    alert("发送失败，有可能消息太长，@用户过多，或是微博服务器繁忙");
                    $("#share_text").val(record);
                }
            },{
               uid :user.id,
               source:appkey,
               status:encodeURIComponent($("#share_text").val()),
               lat:$("#lat").html(),
               long:$("#lng").html()
            },{
                method: 'post'
            });
```

```
                    });
        }else{
            var pic = $("#thump").attr("src").replace("bmiddle","mw1024");

                WB2.anyWhere(function(W){
                    W.parseCMD("/statuses/repost.json", function(sResult, bStatus){
                        if(bStatus == true) {
                            alert("发送成功");
                            $("#dialog-users").dialog( "close" );

                        }
                        else{
                            alert(sResult.error);
                             alert("发送失败，有可能消息太长，@用户过多，或是微博
服务器繁忙");
                            $("#share_text").val(record);
                        }
                    },{
                        id :$("#share_text").attr("name"),
                        source:appkey,
                        status:encodeURIComponent(reposttatus)
                    },{
                        method: 'post'
                    });
                    });
```

```
            }
            */
    },
      startfilter: function(gender){
        var sex = 4;
        switch(gender){
            case 'm':
                sex = 2;
                break;
            case 'f':
                sex = 1;
                 break;
            case 'n':
               sex = 3;
               break;
            default:
              break;
        }
        var gender_condition = $("#sex1 ").get(0).selectedIndex;//0:不限 1:女  2：男
        if(gender_condition == 0){
            return true;
        }else if(sex == gender_condition){
            return true;
        }else{
            return false;
```

```
    }









  },
    showaccuratedetail : function(curPageIndexIndex){
          var html = '';
          var baseurl = 'http://weibo.com/u/';
        var lat = parseFloat($("#lat").html());
        var lng = parseFloat($("#lng").html());
        var r = parseInt($("#radius").html());
        var sort = $("#sort").get(0).selectedIndex;

        document.getElementById("currentpageall").checked = false;

        var is_iPd = navigator.userAgent.match(/(iPad|iPod|iPhone)/i) != null;
        var is_mobi = navigator.userAgent.toLowerCase().match(/(ipod|iphone|android|coolpad|mmp|smartphone|midp|wap|xoom|symbian|j2me|blackberry|win ce)/i) != null;
        if(is_iPd || is_mobi)
        {
            baseurl = 'http://m.weibo.cn/u/';
        }
        $(".friendtabdiv").html('<img src="/img/ref.gif"
```

```
style="margin-left:auto;margin-right:auto"\>');

        WB2.anyWhere(function(W){

             W.parseCMD("/place/nearby/users.json", function(sResult, bStatus){

                  if(bStatus == true) {

                       if(sResult!=''){

                     var users = sResult.users;

                     $("#areacount").html(sResult.total_number);

                          for(var i = 0; i<users.length; i++)

                          {

                              if(itisnavi.prototype.startfilter( users[i].gender)){

                                    var url = baseurl + users[i].id;

                                   html=   html+

                                   '<div  href="javascript:;"   id='+users[i].name+'1   style="
   border:1px                                                                      solid
#C8C8C8;text-align:left;width:91px;height:34px;float:left;display:block;margin:2px;backgrou
nd:#f9f9f9;-moz-border-radius: 4px;-webkit-border-radius: 4px;border-radius:4px;">'+


                                        '<img
style="border:0px;margin:2px;height:30px;width:30px;float:left"
src="'+users[i].profile_image_url+'"/>'+

                                        '<a          href='+url+'          style="border:0px;
color:#000;font-size:13px;width:50px;height:30px;overflow:hidden;word-break:break-all;disp
lay:block;text-align:left;" target="_blank">'+users[i].name +'</a>'+

                                   '</div>';

                              }

                          }

                          $("#areacount").html(sResult.total_number);
```

```
                if(curPageIndexIndex*50 >= sResult.total_number)
                {
                        this.pagiFlag = false;
                }
                else
                {
                        this.pagiFlag = true;
                }
          //new Myright().showacurate(sResult.users);
        }
          else{
              this.pagiFlag = false;
              $("#areacount").html('0');
              html='<font style="font-size:30px;color:red;">已是最后一页</font>';
          }
           $(".friendtabdiv").html(html);
           itisnavi.prototype.recorddisplay();

$(".friendtabdiv").children("div").children("img").bind("click",itisnavi.prototype.friendadd);
        }
    },{
        source:1172770736,
        count:50,
          page:curPageIndexIndex,
        lat:itisnavi.prototype.context.circle.getCenter().lat(),
```

```
                long:itisnavi.prototype.context.circle.getCenter().lng(),
                range:itisnavi.prototype.context.circle.getRadius(),
                starttime:new Date().getTime()/1000-7*24*60*60 ,
                endtime:new Date().getTime()/1000,
                sort:sort

        },{
                method: 'get'
        });
        });

},
darwDialog : function(obj,left ,top,width,height){
        //height = height.substr(0,height.length - 2);
        obj.dialog({
            autoOpen: true,
            height: height,
            width: width,
            modal: true,
             resizable:false,
             draggable:false,
             position:[top,left],
            buttons: {
            },
            close: function() {
```

```
                obj.dialog("destroy");
                obj.remove();
            }
        });
    },
    friendadd : function(e){
            // alert("click");
             var obj = $(e.currentTarget).parent();
            var rgb = $(obj).css('background-color');
            if(rgb.substr(0,3) =='rgb')
            {
                rgb = rgb.match(/^rgb\((\d+),\s*(\d+),\s*(\d+)\)$/);
                function hex(x) {return ("0" + parseInt(x).toString(16)).slice(-2);}
                rgb= "#" + hex(rgb[1]) + hex(rgb[2]) + hex(rgb[3]);
            }
            if(rgb == "#f9f9f9")
            {
                if(parseInt($("#selectcount").html())+1                    <=
parseInt( $("#totalcount").html())){
                    $(obj).css({background:"#548ABE"});
                    itisnavi.prototype.add($(obj).children("a").html())
                }else{
                     alert("到达上限");

                }
```

```
                }
                else
                {

                    $(obj).css({background:"#f9f9f9"});
                     itisnavi.prototype.remove($(obj).children("a").html());
                }
        },
        add : function(name)
        {

                        var len = itisnavi.prototype.users.length;
            for(var i=0; i<len; i++){
                if(itisnavi.prototype.users[i] == name){
                    return;
                }
            }
            var html = '<p id='+name+'><span>'+name+'</span><img
src="/yifenr/img/leftdel.jpg" \></p>';
            $("#friendstotal").append(html);
            itisnavi.prototype.users.push(name);
            $("#selectcount").html( parseInt($("#selectcount").html())+1);

$("#reservecount").html(parseInt( $("#totalcount").html())-parseInt($("#selectcount").html()));
            $("#"+name+" img").bind("click",this.unselect);
```

```
},
showArray : function(){

        for(var i = 0; i < itisnavi.prototype.users.length; i++){
            var name = itisnavi.prototype.users[i];
            var html = '<p id='+name+'><span>'+name+'</span><img src="/yifenr/img/leftdel.jpg" \></p>';
            $("#friendstotal").append(html);
            $("#"+name+" img").bind("click",this.unselect);

        }

        itisnavi.prototype.recorddisplay();
},
removeArrayByName : function(name){
    for(var i=0; i<itisnavi.prototype.users.length; i++){
        if(name == itisnavi.prototype.users[i]){
            itisnavi.prototype.users.splice(i,1);
            return true;
        }
    }
    return false;

},
```

```
remove : function(name)
{

    if(itisnavi.prototype.removeArrayByName(name)){
        $("#selectcount").html( parseInt($("#selectcount").html())-1);
$("#reservecount").html(parseInt( $("#totalcount").html())-parseInt($("#selectcount").html()));

    }

    $("#"+name).remove();
},
unselect : function(e)
{

    itisnavi.prototype.acunumber--;
    if(itisnavi.prototype.acunumber < 1)
    {
        $("#selectokbut").attr("disabled",true);
    }
    var obj = e.currentTarget;
    var removeid = $(obj).parent().attr("id")+'1';
    $("#"+removeid).css({background:"#f9f9f9"});

    var name =    $(obj).parent().children("span").html();
    if(itisnavi.prototype.removeArrayByName(name)){
```

```
                    $("#selectcount").html( parseInt($("#selectcount").html())-1);

        $("#reservecount").html(parseInt( $("#totalcount").html())-parseInt($("#selectcount").html()));

                }
                $(obj).parent().remove();

            },
            recorddisplay : function()
            {
                $("#friendstotal").children("p").each(function(i,n){
                    var obj = $(n).attr("id")+'1';
                    $("#"+obj).css({background:"#548ABE"});
                })
            }

}
```

2 Consistency checking C#.net code:

```
bool consistencychecking(DevComponents.DotNetBar.PanelEx start_panel, List<DevComponents.DotNetBar.PanelEx>
 path, List<string> path_element_name, float temp_length, string temp_length_string,bool is_meets_0)
        {
            foreach (var aaa in Lines)
            {
```

```
            if (aaa.start_panel == start_panel)
            {
                Rights bbb = temp_right.Find(delegate(Rights line) { return ((line.self_panel ==
aaa.End_panel)); });

                if ((aaa.element_length == 0) && (is_meets_0 == true))
                {
                    aaa.is_red = true;
                    is_meets_0_connect = true;
                    is_meets_1_connect = true;
                    return (false);
                }
                else if (aaa.element_length == 0) is_meets_0 = true;
                else is_meets_0 = false;

                if (bbb.has_been_view == true)
                {
                    aaa.is_red = true;
                    return (false);
                }
                else if (bbb.has_been_view_one_time == true)
                {
                    stop_more_deep++;
                    if (stop_more_deep == 2)
                    {
                        stop_more_deep = 0;
                        return (true);
                    }
                }
                else
                {
                    bbb.has_been_view = true;
                    bbb.has_been_view_one_time = true;
                }
```

```
                    if ((temp_length_string != null) && (temp_length_string != ""))
                    { if ((aaa.string_length != null) && (aaa.string_length != "")) temp_length_string
+=aaa.element_name; }
                    else { if ((aaa.string_length != null) && (aaa.string_length != "")) temp_length_string +=
aaa.element_name; }

                    if (aaa.element_length > 0) temp_length += aaa.element_length;

                    path.Add(aaa.End_panel);
                    path_element_name.Add(aaa.element_name);
                    bbb.path.Add(path.ToList());
                    bbb.matrix_string_float.Add(temp_length_string);
                    bbb.matrix_length.Add(temp_length);

                    for(int i= 0 ; i < bbb.matrix_length.Count;i++)
                    {
                        if (bbb.matrix_length[i] < temp_length)
                        {
                            if (temp_length_string.Contains(bbb.matrix_string_float[i]))
                            {
                                first = i;
                                abcd = bbb;
                                aaa.is_red = true;
                                return (false);
                            }

                        }
                        else if (bbb.matrix_length[i] > temp_length)
                        {
                            if(bbb.matrix_string_float[i].Contains(temp_length_string))
                            {
                                first = i;
                                abcd = bbb;
                                aaa.is_red = true;
```

```
                return (false);
            }

        }
    }

    if (aaa.element_length == 0) is_meets_0 = true;

    bool result = consistencychecking(aaa.End_panel, path, path_element_name, temp_length,
temp_length_string, is_meets_0);

    path.Remove(aaa.End_panel);

    if (!result)
    {
        if ((is_meets_0_connect)&&(is_meets_1_connect))
        {
            is_meets_0_connect = false;
            aaa.is_red = true;
            return (false);
        }
        else if (is_meets_1_connect)
        {
            return (false);
        }
        else
        {
            aaa.is_red = true;
            return (false);
        }
    }

    is_meets_0 = false;
```

```
                bbb.has_been_view = false;
                if ((temp_length_string != null) && (temp_length_string != ""))
                    if ((aaa.string_length != null) && (aaa.string_length != ""))
                 {
                     if (temp_length_string.Contains("+"))
                         temp_length_string = temp_length_string.Replace("+" + aaa.element_name, "");
                     else temp_length_string = temp_length_string.Replace(aaa.element_name, "");
                 }
                    else { if ((aaa.string_length != null) && (aaa.string_length != "")) temp_length_string
= temp_length_string.Replace(aaa.element_name, ""); }

                if (aaa.element_length > 0) temp_length -= aaa.element_length;



            }
        }



        return (true);
    }

    void LineMover_MouseUp_IN(object sender, MouseEventArgs e)
    {

    }




    void LineMover_MouseDown_IN(object sender, MouseEventArgs e)
    {

        RefreshLineSelection_IN(e.Location);

    }
```

```
        void LineMover_MouseMove_IN(object sender, MouseEventArgs e)
        {
            if (Moving != null)
            {
                Moving.Line.StartPoint = new PointF(Moving.StartLinePoint.X + e.X -
Moving.StartMoveMousePoint.X, Moving.StartLinePoint.Y + e.Y - Moving.StartMoveMousePoint.Y);
                Moving.Line.EndPoint = new PointF(Moving.EndLinePoint.X + e.X - Moving.StartMoveMousePoint.X,
Moving.EndLinePoint.Y + e.Y - Moving.StartMoveMousePoint.Y);

            }
            RefreshLineSelection_IN(e.Location);
        }

        private void RefreshLineSelection_IN(Point point)
        {
            var selectedLine = FindLineByPoint_IN(Lines_IN, point);

            if (selectedLine != this.SelectedLine)
            {
                this.SelectedLine = selectedLine;
                this.Invalidate();
            }
            if (Moving != null)
                this.Invalidate();

            this.Cursor =
                Moving != null ? Cursors.Hand :
                SelectedLine != null ? Cursors.SizeAll :
                  Cursors.Default;


            if (selectedLine != null)
```

```
            {
                string toolTipinfo;

                if (SelectedLine.element_length > 0)
                    toolTipinfo = "Name:" + SelectedLine.element_name + "\n" + "Duration:" +
SelectedLine.element_length;
                else toolTipinfo = "Name:" + SelectedLine.element_name;

                if(SelectedLine.element_description!=null)
                if (SelectedLine.element_description.Length > 0) toolTipinfo += "\n" + "Description:" +
SelectedLine.element_description;




                this.toolTip1.SetToolTip(this, toolTipinfo);
            }

        }

        static GraphLine FindLineByPoint_IN(List<GraphLine> lines, Point p)
        {
            var size = 10;
            var buffer = new Bitmap(size * 2, size * 2);
            Font afont = new Font("Arial", 15f);

            StringFormat sf = new StringFormat();


            foreach (var line in lines)
            {
                using (var g = Graphics.FromImage(buffer))
                {
                    RectangleF rect = new RectangleF(line.StartPoint.X - p.X + size, line.StartPoint.Y - p.Y
+ size, 0, afont.Height);
```

```
                    g.Clear(Color.Black);
                    g.DrawLine(new Pen(Color.Green, 3), line.StartPoint.X - p.X + size, line.StartPoint.Y - p.Y
+ size, line.EndPoint.X - p.X + size, line.EndPoint.Y - p.Y + size);
                    g.DrawString(line.information, afont, Brushes.Black, rect, sf);

                }

                if (buffer.GetPixel(size, size).ToArgb() != Color.Black.ToArgb())
                    return line;
            }
            return null;
        }


        public class GraphLine
        {
            public GraphLine(float x1, float y1, float x2, float y2, DevComponents.DotNetBar.PanelEx start_panel,
DevComponents.DotNetBar.PanelEx End_panel)
            {
                this.StartPoint = new PointF(x1, y1);
                this.EndPoint = new PointF(x2, y2);
                this.start_panel = start_panel;
                this.End_panel = End_panel;
            }

            public GraphLine()
            {

            }
            public PointF StartPoint;
            public PointF EndPoint;
            public string information;
            public string And_or;
            public string carry_level;
```

```
    public DevComponents.DotNetBar.PanelEx start_panel = new DevComponents.DotNetBar.PanelEx();
    public DevComponents.DotNetBar.PanelEx End_panel = new DevComponents.DotNetBar.PanelEx();

    public string element_name = "";
    public float element_length = -1;
    public string element_description = "";

    public bool huxian = false;

    public bool is_red = false;

    public string string_length = null;

}

GraphLine SelectedLine = null;

MoveInfo Moving = null;

public class MoveInfo
{
    public GraphLine Line;
    public PointF StartLinePoint;
    public PointF EndLinePoint;
    public Point StartMoveMousePoint;
}

public class Rights
{
    Rights()
    {

    }
```

```
        public Rights(DevComponents.DotNetBar.PanelEx start_panel)
        {
            self_panel = start_panel;
            path = new List<List<PanelEx>>();
            has_been_view = false;
        }

        public DevComponents.DotNetBar.PanelEx self_panel = null;
        public List<List<DevComponents.DotNetBar.PanelEx>> path = new List<List<PanelEx>>();
        public bool has_been_view = false;
        public bool has_been_view_one_time = false;
        public List<float> matrix_length = new List<float>();
        public List<string> matrix_string_float = new List<string>();
    }

    public List<GraphLine> Lines_IN = new List<GraphLine>();



    string temp_for_graph_Description = null;
    float temp_for_graph_duration = -1;
    string temp_for_graph_name = null;
    bool temp_for_red = false;

    bool hua_huxian = false;

    DevComponents.DotNetBar.PanelEx[] Panelgroup = new DevComponents.DotNetBar.PanelEx[2];//在¨²panel
内¨²部?用®?于®¨²连¢?接¨®两¢?个?element

    private void connecting_in_panel()
    {
        GraphLine line;

        line = new GraphLine(Panelgroup[0].Location.X, Panelgroup[0].Location.Y,
Panelgroup[1].Location.X,
```

```
        Panelgroup[1].Location.Y, Panelgroup[0], Panelgroup[1]);

    line.element_name = temp_for_graph_name;
    line.element_length = temp_for_graph_duration;
    line.element_description = temp_for_graph_Description;
    line.huxian = hua_huxian;

    Lines_IN.Add(line);

}

int panel_number = 0;
DevComponents.DotNetBar.PanelEx delete_temp = new DevComponents.DotNetBar.PanelEx();

public void panel_MouseDown(object sender, MouseEventArgs e)
{

    mouse_offset = e.Location;
    if (e.Button == MouseButtons.Right)
    {
        this.ctrl = (Control)sender;t
        delete_temp = (DevComponents.DotNetBar.PanelEx)sender;
    }

}

Point mouse_offset;

public void panel_MouseMove(object sender, MouseEventArgs e)
{

    if (e.Button == MouseButtons.Left)
    {
```

```
        int left = PointToClient(Control.MousePosition).X - mouse_offset.X;
        int top = PointToClient(Control.MousePosition).Y - mouse_offset.Y;
        if (left < 0)
            left = 0;
        if (left > this.Width)
            left = this.Width - ((Panel)sender).Width;


        if (top < 0)
            top = 0;
        if (top > this.Height)
            top = this.Height - ((Panel)sender).Height;


        ((Panel)sender).Left = left;
        ((Panel)sender).Top = top;

    }
}

public void panel1_MouseUp(object sender, MouseEventArgs e)
{
    this.Invalidate();
}
public void panel1_Mouseleave(object sender, EventArgs e)
{
    this.Cursor = Cursors.Arrow;

}

private void painting(New_version.meets_table start, Graphics g, int control)
{
    Panelgroup[0] = null;
    Panelgroup[1] = null;
```

```
            temp_for_graph_Description = null;
            temp_for_graph_duration = -1;
            temp_for_graph_name = null;
            hua_huxian = false;

            if (control == 1) hua_huxian = true;
            else hua_huxian = false;

            if (start.is_red) temp_for_red = true;
            else temp_for_red = false;



            for (int i = 0; i < Lines_IN.Count; i++)
            {
                if ((Lines_IN[i].start_panel.Location.X == start.start_x) &&
(Lines_IN[i].start_panel.Location.Y == start.strat_y))
                    Panelgroup[0] = Lines_IN[i].start_panel;



                if ((Lines_IN[i].End_panel.Location.X == start.start_x) && (Lines_IN[i].End_panel.Location.Y
== start.strat_y))
                    Panelgroup[0] = Lines_IN[i].End_panel;
            }

            for (int i = 0; i < Lines_IN.Count; i++)
            {
                if ((Lines_IN[i].start_panel.Location.X == start.end_x) &&
(Lines_IN[i].start_panel.Location.Y == start.end_y))
                    Panelgroup[1] = Lines_IN[i].start_panel;



                if ((Lines_IN[i].End_panel.Location.X == start.end_x) && (Lines_IN[i].End_panel.Location.Y ==
start.end_y))
```

```
                Panelgroup[1] = Lines_IN[i].End_panel;
            }

            if ((Panelgroup[0] == null) && (Panelgroup[1] == null))
            {
                #region 都?不?存ä?在¨²
                DevComponents.DotNetBar.PanelEx panel1 = new DevComponents.DotNetBar.PanelEx();

                this.Controls.Add(panel1);
                panel1.Parent = this;
                panel1.Name = "panel_RESULT" + panel_number.ToString();
                System.Drawing.Point p = new Point(start.start_x, start.strat_y);
                panel1.Location = p;
                panel1.Text = "";
                panel1.Style.BackgroundImage = Image.FromFile(Application.StartupPath +
"\\GRAPH_POINT.fw.png");

                panel1.Width = 30;
                panel1.Height = 30;
                panel1.MouseDown += new MouseEventHandler(panel_MouseDown);
                panel1.MouseMove += new MouseEventHandler(panel_MouseMove);
                panel1.MouseUp += new MouseEventHandler(panel1_MouseUp);
                panel1.MouseLeave += new EventHandler(panel1_Mouseleave);
                panel_number++;


                DevComponents.DotNetBar.PanelEx panel2 = new DevComponents.DotNetBar.PanelEx();

                this.Controls.Add(panel2);
                panel2.Parent = this;
                panel2.Name = "panel_RESULT" + panel_number.ToString();
                System.Drawing.Point p1 = new Point(start.end_x, start.end_y);
                panel2.Location = p1;
                panel2.Text = "";
```

```
                panel2.Style.BackgroundImage = Image.FromFile(Application.StartupPath +
"\\GRAPH_POINT.fw.png");

                panel2.Width = 30;
                panel2.Height = 30;
                panel2.MouseDown += new MouseEventHandler(panel_MouseDown);
                panel2.MouseMove += new MouseEventHandler(panel_MouseMove);
                panel2.MouseUp += new MouseEventHandler(panel1_MouseUp);
                panel2.MouseLeave += new EventHandler(panel1_Mouseleave);

                panel_number++;



                Panelgroup[0] = panel1;
                Panelgroup[1] = panel2;

                temp_for_graph_name = start.element_name;
                temp_for_graph_Description = start.description;
                temp_for_graph_duration = start.length;

                connecting_in_panel();
                #endregion
            }
            else if ((Panelgroup[0] != null) && (Panelgroup[1] == null))
            {
                #region


                DevComponents.DotNetBar.PanelEx panel2 = new DevComponents.DotNetBar.PanelEx();

                this.Controls.Add(panel2);
                panel2.Parent = this;
                panel2.Name = "panel_RESULT" + panel_number.ToString();
```

```
                System.Drawing.Point p1 = new Point(start.end_x, start.end_y);
                panel2.Location = p1;
                panel2.Text = "";
                panel2.Style.BackgroundImage = Image.FromFile(Application.StartupPath +
"\\GRAPH_POINT.fw.png");

                panel2.Width = 30;
                panel2.Height = 30;
                panel2.MouseDown += new MouseEventHandler(panel_MouseDown);
                panel2.MouseMove += new MouseEventHandler(panel_MouseMove);
                panel2.MouseUp += new MouseEventHandler(panel1_MouseUp);
                panel2.MouseLeave += new EventHandler(panel1_Mouseleave);

                panel_number++;

                Panelgroup[1] = panel2;

                temp_for_graph_name = start.element_name;
                temp_for_graph_Description = start.description;
                temp_for_graph_duration = start.length;

                connecting_in_panel();
                #endregion
            }
            else if ((Panelgroup[0] != null) && (Panelgroup[1] != null))
            {
                #region

                temp_for_graph_name = start.element_name;
                temp_for_graph_Description = start.description;
                temp_for_graph_duration = start.length;

                connecting_in_panel();
                #endregion
```

```
        }
        else if ((Panelgroup[0] == null) && (Panelgroup[1] != null))
        {
            #region
            DevComponents.DotNetBar.PanelEx panel1 = new DevComponents.DotNetBar.PanelEx();

            this.Controls.Add(panel1);
            panel1.Parent = this;
            panel1.Name = "panel_RESULT" + panel_number.ToString();
            System.Drawing.Point p = new Point(start.start_x, start.strat_y);
            panel1.Location = p;
            panel1.Text = "";
            panel1.Style.BackgroundImage = Image.FromFile(Application.StartupPath +
"\\GRAPH_POINT.fw.png");

            panel1.Width = 30;
            panel1.Height = 30;
            panel1.MouseDown += new MouseEventHandler(panel_MouseDown);
            panel1.MouseMove += new MouseEventHandler(panel_MouseMove);
            panel1.MouseUp += new MouseEventHandler(panel1_MouseUp);
            panel1.MouseLeave += new EventHandler(panel1_Mouseleave);
            panel_number++;

            Panelgroup[0] = panel1;

            temp_for_graph_name = start.element_name;
            temp_for_graph_Description = start.description;
            temp_for_graph_duration = start.length;

            connecting_in_panel();
            #endregion
        }
```

```
}

private void Digraph_Paint(object sender, PaintEventArgs e)
{
    e.Graphics.InterpolationMode = System.Drawing.Drawing2D.InterpolationMode.High;
    e.Graphics.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.HighQuality;

    GraphicsPath hPath = new GraphicsPath();
    Pen p = new Pen(Color.Black, 1);


    hPath.AddLine(new Point(0, 0), new Point(0, -10));
    hPath.AddLine(new Point(0, 0), new Point(6, -10));
    hPath.AddLine(new Point(0, 0), new Point(-6, -10));
    CustomLineCap HookCap = new CustomLineCap(null, hPath);
    HookCap.WidthScale = 2f;


    p.CustomEndCap = HookCap;

    StringFormat sf = new StringFormat();



    foreach (var line1 in Lines_IN)
    {
        if (line1.start_panel.Location.X >= line1.End_panel.Location.X)
        {
            line1.StartPoint.X = line1.start_panel.Location.X;
            line1.StartPoint.Y = line1.start_panel.Location.Y + line1.start_panel.Height / 2;

            line1.EndPoint.X = line1.End_panel.Location.X + line1.End_panel.Width;
            line1.EndPoint.Y = line1.End_panel.Location.Y + line1.End_panel.Height / 2;
        }
```

```
else
{
    line1.StartPoint.X = line1.start_panel.Location.X + line1.start_panel.Width;
    line1.StartPoint.Y = line1.start_panel.Location.Y + line1.start_panel.Height / 2;

    line1.EndPoint.X = line1.End_panel.Location.X;
    line1.EndPoint.Y = line1.End_panel.Location.Y + line1.End_panel.Height / 2;

}

Point point1 = new Point((int)line1.StartPoint.X, (int)line1.StartPoint.Y);
Point point2 = new Point((int)(line1.StartPoint.X + (int)line1.EndPoint.X) / 2,
((int)line1.StartPoint.Y + (int)line1.EndPoint.Y) / 2);
Point point3 = new Point((int)(line1.StartPoint.X + (int)line1.EndPoint.X) / 2,
((int)line1.StartPoint.Y + (int)line1.EndPoint.Y) / 2 - 50);
Point point4 = new Point((int)line1.EndPoint.X, (int)line1.EndPoint.Y);
Point[] myPoints = { point1, point2, point4 };
Point[] myPoints_1 = { point1, point3, point4 };

if (line1.is_red) p = new Pen(Color.Red, 1);
else p = new Pen(Color.Black, 1);
p.CustomEndCap = HookCap;

Font afont2 = new Font("Arial", 12f);
Font afont = new Font("Arial", 8f);

Rectangle rect2 = new Rectangle(((int)line1.StartPoint.X + (int)line1.EndPoint.X) / 2 - 25,
((int)line1.StartPoint.Y + (int)line1.EndPoint.Y) / 2 - 20, 0, afont2.Height);
Rectangle rect3 = new Rectangle(((int)line1.StartPoint.X + (int)line1.EndPoint.X) / 2 - 22 +
8, ((int)line1.StartPoint.Y + (int)line1.EndPoint.Y) / 2 - 14, 0, afont2.Height);

Rectangle rect4 = new Rectangle(((int)line1.StartPoint.X + (int)line1.EndPoint.X) / 2 - 25,
((int)line1.StartPoint.Y + (int)line1.EndPoint.Y) / 2 - 70, 0, afont2.Height);
Rectangle rect5 = new Rectangle(((int)line1.StartPoint.X + (int)line1.EndPoint.X) / 2 - 22 +
```

```
8, ((int)line1.StartPoint.Y + (int)line1.EndPoint.Y) / 2 - 64, 0, afont2.Height);

                string temp_draw = line1.element_name.Substring(1, line1.element_name.Length - 1);

                if (line1.element_name.Contains("Tnp"))
                {
                    foreach (var line2 in Lines_IN)
                    {
                        if (line2.start_panel == line1.End_panel)
                        {
                            line1.start_panel.Visible = false;

                        }

                        if (line2.End_panel == line1.start_panel)
                        {

                            line1.End_panel.Visible = false;
                        }
                    }

                }
                else
                {
                    if (line1.huxian)
                        e.Graphics.DrawCurve(p, myPoints_1, 0.80F);
                    else e.Graphics.DrawCurve(p, myPoints, 0.80F);

                    if (!line1.huxian)
                    {
                        if (line1.element_length >= 0)
                        {
                            e.Graphics.DrawString("T", afont2, Brushes.Black, rect2, sf);
                            e.Graphics.DrawString(temp_draw + "(" + line1.element_length + ")", afont,
```

```
Brushes.Black, rect3, sf);
                        }
                        else
                        {
                            e.Graphics.DrawString("T", afont2, Brushes.Black, rect2, sf);
                            e.Graphics.DrawString(temp_draw, afont, Brushes.Black, rect3, sf);
                        }
                    }
                    else
                    {
                        if (line1.element_length >= 0)
                        {
                            e.Graphics.DrawString("T", afont2, Brushes.Black, rect4, sf);
                            e.Graphics.DrawString(temp_draw + "(" + line1.element_length + ")", afont,
Brushes.Black, rect5, sf);
                        }
                        else
                        {
                            e.Graphics.DrawString("T", afont2, Brushes.Black, rect4, sf);
                            e.Graphics.DrawString(temp_draw, afont, Brushes.Black, rect5, sf);
                        }

                    }
                }
            }
        }

    }

}
```

APPENDIX B

Speech Tags

POS Tag	Description	Example
CC	coordinating conjunction	and
CD	cardinal number	1, third
DT	determiner	the
EX	existential there	there is
FW	foreign word	d'hoevre
IN	preposition/subordinating conjunction	in, of, like
JJ	adjective	big
JJR	adjective, comparative	bigger

JJS	adjective, superlative	biggest
LS	list marker	1)
MD	modal	could, will
NN	noun, singular or mass	door
NNS	noun plural	doors
NNP	proper noun, singular	John
NNPS	proper noun, plural	Vikings
PDT	predeterminer	both the boys
POS	possessive ending	friend‛s
PRP	personal pronoun	I, he, it
PRP$	possessive pronoun	my, his
RB	adverb	however, usually, naturally, here,

		good
RBR	adverb, comparative	better
RBS	adverb, superlative	best
RP	particle	give up
TO	to	to go, to him
UH	interjection	uhhuhhuhh
VB	verb, base form	take
VBD	verb, past tense	took
VBG	verb, gerund/present participle	taking
VBN	verb, past participle	taken
VBP	verb, sing. present, non-3d	take
VBZ	verb, 3rd person sing. present	takes

WDT	wh-determiner	which
WP	wh-pronoun	who, what
WP$	possessive wh-pronoun	whose
WRB	wh-abverb	where, when