

University of Warwick institutional repository: <http://go.warwick.ac.uk/wrap>

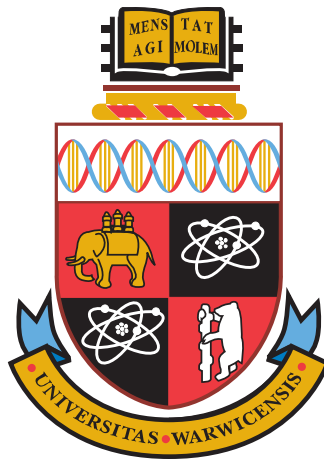
**A Thesis Submitted for the Degree of PhD at the University of Warwick**

<http://go.warwick.ac.uk/wrap/71036>

This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it. Our policy information is available from the repository home page.



# Data Fusion for Human Intelligence and Crisis Management: Handling Information from Untrusted Sources

Syed Sadiqur Rahman  
(BSc, MSc)

*A thesis submitted in partial fulfilment of the requirements for  
the degree of  
Doctor of Philosophy in Engineering*

*School of Engineering  
University of Warwick  
2014*



# Contents

<b>Acknowledgements</b>	<b>xii</b>
<b>Declaration</b>	<b>xiv</b>
<b>Abstract</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research Objectives . . . . .	3
1.2 Main Contributions of the Thesis . . . . .	4
1.3 About TEASE . . . . .	5
1.4 Thesis structure . . . . .	6
<b>2 Managing Crisis Information - State of the Art</b>	<b>11</b>
2.1 Data Structure and Sharing Crisis and Situational Information . . . . .	12
2.2 Use of Open-Source Data, Social Networks and Crowdsourcing in Crisis Management . . . . .	12
2.2.1 Incident Detection and Early Warning . . . . .	13
2.2.2 Situation Awareness . . . . .	14
2.2.3 Trust Evaluation . . . . .	16
2.3 Provenance and Trustworthiness of Information for Crisis Management	18
2.4 Verification of Crowd-sourced Information . . . . .	21
2.4.1 DARPA Network Challenge . . . . .	21
2.4.2 Referral-based Verification . . . . .	22
2.4.3 Ushahidi - Swift River . . . . .	23
2.5 Consistency Evaluation of Information for Crisis Management . . . . .	24
2.6 Summary . . . . .	26
<b>3 Uncertainty in the Wake of Crisis and Emergency</b>	<b>28</b>
3.1 Harvesting Open-source Data from Twitter . . . . .	29
3.1.1 Collecting Tweets from Targeted User Accounts . . . . .	29
3.1.2 Data Collection using Twitter Search API . . . . .	31
3.2 Reproducing a Graphical Timeline of Incident-related Tweets . . . . .	31
3.3 Uncertainty in the Crowds during Mumbai Incident 2008 . . . . .	32
3.3.1 Inconsistent and Contradictory Information . . . . .	33

3.3.2	Hoax and Rumour . . . . .	33
3.4	Tool Support: Finding Popular Users by Analysing Retweets and Mentions . . . . .	36
3.5	Uncertainty in the Aftermath of Haiti Earthquake in 2010 . . . . .	37
3.5.1	Implicit Disclosure of (Location) Information . . . . .	38
3.5.2	Windows of Uncertainty . . . . .	40
3.6	Summary . . . . .	41
<b>4</b>	<b>Provenance and Quality Factors Affecting Trust in Information</b>	<b>43</b>
4.1	Provenance Factors Affecting Trust in Information . . . . .	44
4.1.1	Identity of Informer . . . . .	45
4.1.2	Location of Informer . . . . .	46
4.1.3	Reputation . . . . .	47
4.1.4	Popularity . . . . .	47
4.1.5	Context/situation, Interest and Ethics . . . . .	48
4.1.6	Social Relation . . . . .	49
4.1.7	Corroboration . . . . .	49
4.1.8	Competence . . . . .	50
4.1.9	Conviction/Certainty . . . . .	50
4.2	Information Quality Factors that Influence Trust . . . . .	51
4.2.1	Security of Information . . . . .	51
4.2.2	Freshness of Information . . . . .	51
4.2.3	Correctness and Accuracy . . . . .	52
4.2.4	Completeness . . . . .	52
4.2.5	Objectivity . . . . .	53
4.3	Factors related to Information Consumer that Influence Trust . . . . .	53
4.3.1	Disposition to Trust . . . . .	53
4.3.2	Propensity to Risk Taking . . . . .	55
4.4	Vulnerabilities of the Provenance Quality and Trust Factors . . . . .	55
4.5	Summary . . . . .	55
<b>5</b>	<b>Decision Support System: High-Level Design and Algorithm</b>	<b>58</b>
5.1	Top-Level Design and Architecture . . . . .	59
5.1.1	Information Source Filter . . . . .	59
5.1.2	Tactical Situation Object (TSO) Encoder . . . . .	61
5.1.3	Scoring Function . . . . .	62
5.1.3.1	Exemplification of Scoring Function . . . . .	64
5.1.4	Consistency Analysis and Conflict Resolution . . . . .	65
5.1.5	Combiner Function: Calculating an Overall Total Score of a World View . . . . .	66
5.1.5.1	Comparative Analysis of Different Average Functions	68
5.1.5.2	Combiner Function . . . . .	71
5.1.6	Decision Making Policy . . . . .	73
5.2	Summary . . . . .	74

<b>6</b>	<b>Structuring Data into Suitable Format</b>	<b>76</b>
6.1	Tactical Situation Object (TSO) . . . . .	76
6.1.1	Structure of a Tactical Situation Object (TSO) . . . . .	77
6.2	Usability Issues of the TSO Data Dictionary . . . . .	81
6.3	Constructing TSO Schema from TSO Message Structure and its Data Dictionary . . . . .	81
6.3.1	Adding new Elements to the Existing Schema . . . . .	85
6.4	Modification and Extension of TSO . . . . .	86
6.5	Tactical Situation Object (TSO) Encoding . . . . .	90
6.6	Summary . . . . .	90
<b>7</b>	<b>Methodology of Constructing World Views</b>	<b>92</b>
7.1	Clustering TSO based on Event Location, Event Type, Actors and Sentiment/Positivity . . . . .	95
7.2	Modelling and Consistency Checking with Alloy . . . . .	100
7.2.1	Alloy Language . . . . .	101
7.2.2	Alloy Analyzer . . . . .	103
7.2.3	Scope . . . . .	103
7.3	Processing Individual Messages . . . . .	104
7.3.1	Encoding TSO in Alloy . . . . .	105
7.3.2	Internal Consistency Checking . . . . .	108
7.4	Construction of Consistent World Views . . . . .	110
7.4.1	Processing with a minimum scope . . . . .	110
7.4.2	Reprocessing with a larger scope and reliability of world views . . . . .	117
7.4.3	Impact of a new message on existing world views . . . . .	117
7.5	World View Generation: Some Observations . . . . .	120
7.6	Summary . . . . .	122
<b>8</b>	<b>Realising the Decision Support System</b>	<b>124</b>
8.1	Encoding Messages into TSO . . . . .	127
8.2	Viewing TSO Data and Scoring Messages based on an Organisational Policy . . . . .	130
8.3	Encoding TSO into Alloy and Feeding into Alloy Analyzer . . . . .	134
8.4	Construction of Consistent World Views through Alloy Automation . . . . .	135
8.5	User Interface: Presenting an Ordered List of World Views for Decision Makers . . . . .	139
8.5.1	Synchronising Client-side and Server-side Data . . . . .	142
8.6	Application of Decision Making Policy . . . . .	143
8.7	Lines of Code Metrics . . . . .	144
8.8	Summary . . . . .	146
<b>9</b>	<b>Technical Demonstration of the Decision Support System</b>	<b>147</b>
9.1	How to Use the System . . . . .	149
9.1.1	Harvesting Data . . . . .	149
9.1.2	Data Set . . . . .	150

9.1.3	Encoding of Plain Text Messages into TSO . . . . .	151
9.1.4	Assigning Scores against Trust Factors . . . . .	151
9.1.5	Consistency Analysis, World View Generation and Scoring .	152
9.2	Experiments . . . . .	158
9.2.1	Is the Consistency Analysis Performed by DSS Correct? . .	158
9.2.2	Logical Soundness of the World Views . . . . .	159
9.2.3	Impact of a new message on existing world views . . . . .	160
9.2.4	Decision Making Policy - Sensitivity Analysis . . . . .	161
9.2.5	Constructing World Views with a Larger Data Set . . . . .	163
9.2.6	World Views Adhere to the Logical Rules – if a premise is satisfiable, then its consequent is also satisfiable . . . . .	166
9.3	Processing Capacity of Alloy Analyzer and its Suitability in Real Life Cases . . . . .	166
9.4	Summary . . . . .	171
<b>10</b>	<b>Conclusion and Future Work</b>	<b>173</b>
10.1	Consistent Vs Fuzzy World Views: A Critical Analysis . . . . .	175
10.2	Limitations and Future Work . . . . .	179
	<b>References</b>	<b>183</b>
	<b>Glossary</b>	<b>197</b>
	<b>Appendices</b>	<b>202</b>
<b>A</b>	<b>Structure of Tweets Returned by the Twitter APIs</b>	<b>202</b>
A.1	Structure of Tweets Returned by the Twitter User Timeline API . .	202
A.2	Structure of Tweets Returned by the Twitter Search API . . . . .	204
<b>B</b>	<b>Construction of World Views: Input, Process and Output</b>	<b>205</b>
B.1	Input Messages used to Construct World views in Chapter 9 . . . . .	205
B.2	World Views Generated in the Experiment in Chapter 9 . . . . .	210
B.3	World Views Generated with a Smaller Scope . . . . .	213
<b>C</b>	<b>Alloy Representation of the Facts and Messages Used in Consis-</b> <b>tency Analysis</b>	<b>214</b>
<b>D</b>	<b>High Level Components of TSO Schema</b>	<b>230</b>
<b>E</b>	<b>Functions and Stored Procedures used in the Back End</b>	<b>232</b>
E.1	Storing Tweets into Database . . . . .	232
E.2	Scoring World Views . . . . .	234
<b>F</b>	<b>Alloy Automation and World View Generation Source Code</b>	<b>238</b>

# List of Figures

1.1	Near Real-Time Disaster Information of Twitter [125] . . . . .	3
1.2	Decision Support System Architecture . . . . .	8
3.1	Message Timeline . . . . .	31
3.2	Mapping Popularity based on Retweets and Mentions . . . . .	36
3.3	Classification of Tweets based on the Main Topics of Messages . . . . .	39
3.4	Rumour created Uncertainty about Flood for about 19 minutes . . . . .	42
4.1	People’s Response to an Incomplete Information on ‘Top Tax Cheats’ [100] . . . . .	54
5.1	Data Flow in the System Architecture . . . . .	60
5.2	Effect of Mean, Geometric Mean, Harmonic Mean and Root Mean Square . . . . .	70
6.1	A Tactical Situation Object (TSO) Created by the Emergency Services Call Centre [43] . . . . .	77
6.2	A Tactical Situation Object (TSO) Created by Fire Service in Response to the TSO in Figure 6.1 [43] . . . . .	78
6.3	Hierarchy of some of the TSO Elements . . . . .	84
7.1	Data Flow in the System Architecture: An Extended View . . . . .	94
7.2	Worlds in Tree View: Messages are grouped according Location of Incident first . . . . .	96



7.3	Worlds in Tree View: Messages are grouped according to Category of Incident first . . . . .	98
7.4	All Occurrences of a Selected Actor . . . . .	99
7.5	Contradictory Messages in Tree View . . . . .	101
7.6	A File System Model in Alloy . . . . .	102
7.7	Instance of an Alloy Model . . . . .	103
7.8	Specification of Different Types of Objects in Alloy . . . . .	105
7.9	Specification of Other Extended Objects in Alloy . . . . .	106
7.10	Specification of a Car and other Vehicles in Alloy . . . . .	107
7.11	An Alloy Predicate . . . . .	108
7.12	Individual Messages Stored in Database with their Truth Values . .	109
7.13	A Pileup . . . . .	111
7.14	World View Construction Process . . . . .	112
7.15	Process Diagram of the World View Generator (Breadth-First Search)	114
7.16	Combine a New Message with Existing World Views . . . . .	118
8.1	Specifying the TSO Schema for Constructing a TSO . . . . .	127
8.2	An XML file containing only the Root Element of the TSO . . . . .	128
8.3	Creating TSO in Grid View . . . . .	128
8.4	Using the Context Menu to Append a TSO Element . . . . .	129
8.5	Compulsory Child-Elements of a TSO Element are Added Automatically . . . . .	129
8.6	Context Aware Drop-down Menu Makes it Easy to Enter the Correct Values for TSO Elements . . . . .	130
8.7	TSO Scoring System User Interface (View-1) . . . . .	131
8.8	TSO Scoring System User Interface (View-2) . . . . .	132
8.9	TSO Scoring System User Interface (View-3) . . . . .	133
8.10	TSO Scoring System User Interface (View-4) . . . . .	133

8.11	TSO Scoring System User Interface (View-5)	134
8.12	A Sample and Incomplete Alloy Input File	137
8.13	A Sample Alloy Input File after Updating it Programmatically	138
8.14	World Views Before Applying Decision Making Policy	140
8.15	Messages in a Selected World View	141
8.16	World Views After Applying Decision Making Policy	143
9.1	GUI for Harvesting data from a User's Timeline on Twitter	149
9.2	GUI for Harvesting data using Twitter Search API	150
9.3	World Views Generated from the 10 Messages Listed in Table 9.1	157
9.4	World Views Generated by the Decision Support System from the Messages Listed in Table 9.1	158
9.5	Impact of Decision Making Policy on World Views (a)	161
9.6	Impact of Decision Making Policy on World Views (b)	162
9.7	Impact of Decision Making Policy on World Views (c)	162
9.8	World Views Generated from 20 Messages	165
9.9	Change in Score has Reduced Overlapping of the World Views	165
9.10	Time Taken to Generate World Views from 10 Messages in Batch Mode	167
9.11	Time Taken to Generate World Views from 10 Messages in Serial Mode	168
9.12	Time Taken to Generate World Views from 25 Messages in Serial Mode	169
10.1	A Fuzzy World View	177
10.2	World Views: Consistent and Fuzzy, Two-in-One View	178
10.3	Two World Views Stacked on Each Other is Indicated by a Circled Number	182
A.1	Twitter Search API's in Output JSON Format	204
B.1	Process Diagram of the World View Generator (Depth-First Search)	209
D.1	High Level Components of TSO Schema	231

# List of Tables

3.1	Contradictions Found in Tweets about Mumbai Incident . . . . .	34
3.2	<b>mumbaiupdates</b> ' Tweets Before and After the Creation of the Hoax	35
3.3	Coverage of Topics within Tweet Sample Set (Haiti Earthquake, 2010)	38
3.4	People Reveal their Location Implicitly on Twitter . . . . .	39
3.5	Tweets Containing Valuable Information . . . . .	40
3.6	Tweets related to a Hoax that spread after the Earthquake in Haiti	41
4.1	Vulnerability or Limitation of Factors that Influence Trust . . . . .	56
5.1	Scoring Policy for Location . . . . .	64
5.2	Scoring Policy for Freshness or Timeliness . . . . .	64
5.3	Scoring Policy for Reputation . . . . .	65
5.4	Effect of Different Average Functions (a) . . . . .	68
5.5	Effect of Different Average Functions (b) . . . . .	69
5.6	Calculating Composite Score . . . . .	72
5.7	Calculating the Overall Total Score of a World View . . . . .	73
5.8	Change in Overall Total Score of a World View due to Decision Making Policy . . . . .	73
6.1	TSO Message Structure defines some of the Elements and Sub-Elements [43]	82
6.2	Codes defined in TSO Data Dictionary (a) [44] . . . . .	83
6.3	Codes defined in TSO Data Dictionary (b) [44] . . . . .	84

6.4	Some TSO Codes Need Renaming . . . . .	87
8.1	Summary on Different Components of the DSS . . . . .	125
8.2	Message Clusters and World Views in Database . . . . .	139
8.3	Metrics on the Lines of Code written to develop the DSS Framework	145
9.1	Test Data Set-1: The Messages Used to Construct World Views and their Satisfiability Evaluated by the DSS . . . . .	150
9.2	Proximity of User, Age of Information and Scores . . . . .	152
9.3	Test Data Set-1: The Messages Used to Construct World Views and their Satisfiability Evaluated by the DSS . . . . .	154
9.4	Summary of Information Presented in Table 9.1, 9.2 and 9.3 . . . . .	155
9.5	Calculation of the Overall Score of the World View AFGIJ . . . . .	157
9.6	World Views Generated from the 10 Messages Shown in Table 9.1 .	159
9.7	Test Data Set-2: The Messages Used to Construct World Views and their Satisfiability Evaluated by the DSS . . . . .	163
9.8	World Views Generated from the Test Data n Table 9.4 and Table 9.7	164
B.1	Complete Set of Messages Used to Construct World Views and their Satisfiability Evaluated by the DSS . . . . .	205
B.2	World Views Generated from the Test Data Listed in Table 9.4 and Table 9.7 . . . . .	210
B.3	World Views Generated from the Test Data in Table 9.4 and Table 9.7 with only 2 Cars Available in the Universe (Scope) . . . . .	213

# Acknowledgements

I would like to express my gratitude to everyone without whose support it would not be possible for me to come to this stage. Many people have helped me in many different ways at different levels; some have taught me to respect education and educated people, while others have actively encouraged me to pursue the highest level of education.

I am extremely grateful to my supervisors, Professor Sadie Creese and Professor Michael Goldsmith. They are simply great supervisors. They have made my PhD studentship much easier than I initially anticipated. Without my supervisors' academic and moral support, pragmatism and broadness of mind, it would be impossible for me to reach this stage of writing a PhD thesis. I thank Sadie and Michael from the core of my heart.

I express my gratitude to Thales Research and Technology (UK) Ltd, my industrial sponsor, and the Engineering and Physical Sciences Research Council (EPSRC), the UK Government's main agency for funding research in engineering and the physical sciences, for providing me the internship and financial support.

I would like to thank Dr Rachel Craddock, my industrial mentor, Glyn Jones, Darren Price and others from Thales for their valuable discussion and feedback. I would also like to thank Dr David Lund, and George Mourakis of HW Communications Ltd. for their discussions and cooperation during TEASE meetings.

I would also like to thank Dr. Helen Treharne and Dr. Jane Sinclair for their invaluable feedback and for agreeing to be my examiners for this thesis.

A special gratitude goes to all members of the e-Security Group at the University of Warwick and the Cyber Security Group at University of Oxford. These are the names that I cannot forget because of their great contributions (in various ways) and helping hands including proof reeding of this thesis: Dr Jason Nurse, Dr Ioannis Agrafiotis, Dr Jassim Happa, Dr Adedayo Adetoye, Dr Nick Moffat, Dr Duncan Hodges, Adrian Duncan, Dr Nick Papanikolaou and Mike Auty.

I would also like to thank Dr Kevin Neailey, Dr David Preston, Dr Muhammad Shahidur Rahman, Dr Mohammad Reza Selim, Dr Muhammad Azizur Rahman, Mohammad Badrudduza Shah, Dr Mohammed Aboulsamh, Paul Hopkins, Dr Tom Bashford-Rogers, Thomas Gibson-Robinson, Elmedin Selmanovic, Dr Mark Josephs, Dr Jay Bal and Dr Xiao Ma for their valuable advice, encouragement and help.

It will be unfair on my part if I do not express my gratitude to Dr David Power, without whose help learning Alloy would be much harder for me. David always answered my questions and corrected my mistakes. Thank you David.

I would like to thank Mizan Syed (Internet Technical Manager at UK Prime Minister's Office/Cabinet Office) and Syed Muizur Rahman (Lead Software Engineer at Capgemini) for their valuable discussion and finding bugs in my code. They have saved a lot of hair on my head from being pulled out by losing some of theirs! A special thank you goes to my Chowdhury Asif Mohammed Samir for editing some of the figures with Photoshop.

Most importantly, I am deeply and forever indebted to my parents, brothers and sisters for their unbounded love, support and encouragement. It would be impossible for me to achieve anything without their support and guidance. My late father, who has made the greatest sacrifice to educate his children, would be proud and delighted the most at seeing our success. My oldest brother who has always been a great source of inspiration played the role of a great mentor. I thank all of them.

The person who has made the biggest sacrifice during my PhD studentship and always saved me from distractions is my wife Umme Salma Chowdhury. I would like to say a special thank you to her.

Finally, I thank Allah for giving me the ability to do the hard work and the chance to meet all these great people.

# Declaration

The work presented in this thesis is original and has never been submitted for another degree or qualification to any other university or institution of learning. All images, figures and tables have been created by the author of this thesis, unless otherwise stated.

Parts of this thesis have been published in two conference papers. For example, Information Provenance and Quality Factors that Influence Trust in Information have previously appeared and been discussed in [103] and [111]. While the whole of the research work presented in [111] has been performed by myself, the co-authors have some contribution to the work presented in [103]. However, at least 60% of the total research work presented in this paper belongs to me.

Syed Sadiqur Rahman, May 2013

# Abstract

Situation awareness is a key requirement in managing civil contingencies, since major incidents, accidents and natural disasters are by their very nature highly unpredictable and confusing situations. It is important that those responsible for dealing with them have the best available information. The mash-up approach brings together information from multiple public and specialist sources to form a synoptic view, but the controller is still faced with multiple, partial and possibly conflicting reports from untrusted sources. The aim of this research is to investigate how the varying provenance of the data can be tracked and exploited to prioritise the information presented to a busy incident controller, and to synthesise a model or models of the situation that the evidence pertains to.

The approach in this research is to develop a system involving novel approach and techniques to allow incident controllers and similar decision makers to augment official information input streams with information contributed by the wider public (either explicitly submitted to them or harvested from social networks such as Facebook and Twitter), and to be able to handle inconsistencies and uncertainty arising from the unreliability of such sources in a flexible way. The system takes in situational data in a structured format, such as the Tactical Situation Object (TSO) proposed by OASIS, a project funded by the European Framework Programme 6 (FP6) and performs an automated logical consistency checking in order to isolate inconsistent and absurd messages, identify the inconsistency between messages and cluster the consistent messages together. Each cluster of consistent messages that gives a possible view of a situation that the evidence pertains to is referred to as a ‘World View’. The logical consistency checking is performed using Alloy and Alloy Analyzer (sic). Finally, the system presents a set of possible world views, each internally consistent, which are ranked based upon an initial information provenance and quality metric (configured by the user) which is used to score the individual data items. The provenance and quality metric includes those factors that influence trust in information such as identity and location of informant, reputation, corroboration, freshness of information, etc. The result is a set of world views prioritised according to the provenance, trust and information quality metric. This thesis also presents some experimental results as proof of the concept. The experimentation has been carried out with a very small set of data to make the automation (automatic experimentation) feasible. However, a theoretical proof is offered to demonstrate the



viability of the concept. Future work includes testing the system in real-life cases, in order to understand the utility of the system.

**Keywords:** Crisis, Emergency, Situation Awareness, Crowdsourcing, Social Networks, Provenance, Trust, Consistency Analysis, Formal Methods, Alloy, Decision Support

## CHAPTER 1

# Introduction

A crisis and emergency situation is very perplexing and it often requires a prompt decision to be made and an action to be taken for avoiding loss. However, decision making is always a difficult task, especially in the case of crisis and emergency. This is mostly due to the associated uncertainty and one of the main reasons for uncertainty is the lack of reliable information. One of the most important things that helps to make a good decision is the availability of information that reduces uncertainty. When the information comes from reliable and official sources, decision making becomes comparatively easy as the decision makers can make an informed decision based on reliable information. Therefore, traditionally, emergency responders and humanitarian organisations depend or tend to depend on the official data collected from known reliable sources, which is easier to verify [143]. These official sources include trained staff, professional journalists, news outlets, different sensors e.g. CCTV, RADAR, satellite, mobile phone, etc. However, in many cases, especially in a large-scale disaster, information from known and reliable or official sources becomes very scarce; the members of the public, who are neither trusted nor distrusted, may become the main source of information and a huge amount of real-time information typically becomes available on the web, especially on social networks.

Situation awareness is a key requirement in managing civil contingencies and it is important that those responsible for dealing with crisis and emergency situation have the best available information. This is why everyone, starting from individuals to news organisations, from aid workers to government intelligence services and customer services of different companies, are now listening to the crowds on the social networks in order to collect information [82, 111, 45]. The micro-blogging service Twitter alone has a large user base that was attracting 190 million visitors

each month according to a report published in July 2010[117]. Twitter, which used to produce about 65 million messages (tweets) a day in 2010 [117], now generates 500 million tweets everyday [145]. 532m status updates are updated on Facebook, while 172 million different people visit Facebook every day [126].

People share all sorts of information with others on Social networks (e.g. Twitter, Facebook, etc.) and Photo/Video hosting services (e.g. Flickr). When a disaster strikes an area, people of that place share or disseminate information about the disaster from their first hand experience. Thus, social networks receive a large amount of postings during crises and disasters from eye witnesses and others, both from inside and outside the affected locations [24, 109, 125]. As a result, disaster information starts to emerge on social networks almost real-time. Figure 1.1 shows some near real-time information that became available on Twitter immediately after an earthquake in San Francisco in March 2009 [125]. Social networks, thus, turns into a ‘virtual news portal’, which has been witnessed during many incidents including wildfires in California, earthquake in San Francisco, and terrorist attacks in Mumbai [109, 125, 24]. These social network postings might be useful to agencies charged with responding to crisis situations (whether natural disasters or terrorist and other man-made incidents). A synoptic view of a situation may be created for improved situation awareness from such open-source data, by using the mash-up approach that brings together information from multiple public and specialist sources including the social networks. However, in order for such data to have utility, it must be of appropriate quality.

The availability of a large volume of information on the web has given birth to the new concept and practice of ‘crowdsourcing’. Crowdsourcing is the practice of obtaining required information or services by enlisting the services of a number of people, either paid or unpaid, typically via the Internet [32]. However, the problem still remains. The reliability of information harvested through crowdsourcing is unknown. Therefore, the emergency responders and decision makers must use such information with extra caution.

Devising a method of assessing quality, reliability and trustworthiness of information will help utilising the information from social networks. It is this requirement that motivates the research presented in this thesis.

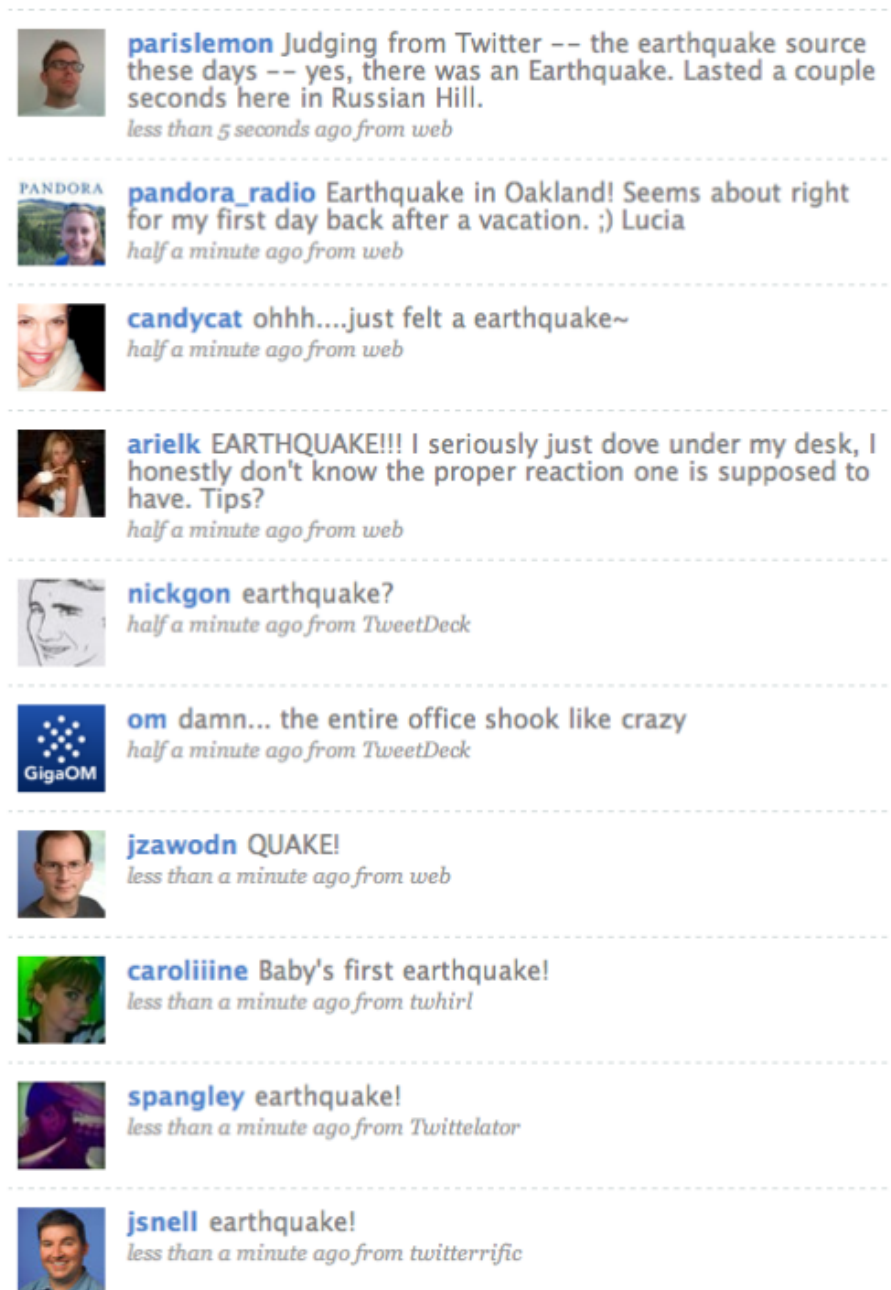


Figure 1.1: Near Real-Time Disaster Information of Twitter [125]

## 1.1 Research Objectives

The main objective of this research is to:

1. Study the factors that influence the trustworthiness of information i.e. the factors that lead people to trust or distrust a piece of information. These factors may be used to prioritise some information in the face of uncertainty.

2. Investigate whether (and possibly how) information from untrusted sources can be used to make informed decision in the case of crisis and emergency.
3. Explore the possibility of using provenance or lineage of information and other factors that influence trust to prioritise information during crisis and emergency that may help emergency responders or decision makers to make decision in the face of uncertainty.
4. Investigate whether logic based automated reasoning can be used to validate or check the consistency of large amount of information during crisis and emergency, which may help realising the full potential of the large volume of information available on the Internet and social networks in particular.

## 1.2 Main Contributions of the Thesis

Decision makers, emergency responders and humanitarian organisations prefer to use information from official sources (e.g. mainstream media outlets and automatic sensors like CCTV, RADAR, satellite, mobile phone, etc.) for making decisions. They are reluctant to use information from untrusted sources (e.g. online social networks) as they are deemed unverifiable and unreliable [143], even though in many cases, especially in a large-scale disaster, the members of the public and online social networks become the main source of information. Although, a significant amount of research has been carried out to improve the situational awareness and emergency response utilising the information from official sources [28, 67, 124, 168, 53, 78], the amount of work done by the research community to utilise open-source information for situation awareness and crisis response is still very limited and insufficient, because they have not addressed the issue of assessing the reliability and trustworthiness of such information. This thesis seeks to address this challenge by assessing the trustworthiness of information in the face of uncertainty based on its provenance-related data.

The novelty of this research stems from the use of provenance (also known as lineage or pedigree), other information quality factors and human trust factors to utilise the power of open-source information. Provenance of information refers to the source of information such as who gave (or produced) the information, the derivation history of information, what data was used to generate it, and also finding the trail of how the information has passed from one source to the other and how it has

been changed. Provenance of information has direct and indirect links with security properties, especially with information authenticity, integrity and access control [80]. Thus, provenance information helps to assess the quality of information (correctness, authenticity, integrity, etc.) and thereby, helps to determine the level of trust that can be attributed to it [56, 140]. A detailed discussion on different factors, including provenance and quality of information, that influence trust in information appears in Chapter 4.

Another novel aspect of this research is the creation of world views from open-source data. Each world view is a possible picture of a situation constructed from the evidence found in the harvested open-source data, and a world view contains only those messages that are consistent with each other. The use of Alloy<sup>1</sup> to perform automated logical consistency-checking in order to detect contradiction between messages is also an innovative approach. Generally, logical reasoning and specifically Alloy have been widely used in various applications. However, the use of Alloy and the Alloy Analyzer (sic) for checking the consistency of messages, especially in the context of crisis and emergency management, is a novel approach. A detailed discussion on this novelty appears in the chapter ‘Managing Crisis Information – State of the Art’ (Chapter 2).

### 1.3 About TEASE

My research and the result of my case studies (partly described in Chapter 3) have inspired the joint collaborative project, TEASE (Trust Enabling Augmented-reality Support for information-Environments<sup>2</sup>) to commence with a broader view and objective. The TEASE project was led by a consortium of two academic and two industrial partners:

1. Department of Computer Science (Cyber Security Group),  
University of Oxford, UK (formerly the e-Security Group in WMG Digital  
Laboratory of University of Warwick) – [www.cybersecurity.ox.ac.uk](http://www.cybersecurity.ox.ac.uk)
2. Department of Psychology, University of Warwick, UK – [www2.warwick.ac.  
uk/fac/sci/psych](http://www2.warwick.ac.uk/fac/sci/psych)
3. Thales Research and Technology (UK), Reading, UK – [www.thalesgroup.  
com/uk](http://www.thalesgroup.com/uk)

---

<sup>1</sup>Alloy is a model specification language based on first-order logic and set theory [92].

<sup>2</sup><http://www.tease-project.info>

4. HW Communications Ltd, Lancaster, UK – [www.hwcomms.com](http://www.hwcomms.com)

The main objectives of the two-year long (from January 2011 to January, 2013) project were to:

- Explore how the provenance of information can be used to provide a measure of confidence that a user should have in information and its source.
- Investigate the best possible ways to present the provenance data and the confidence measure to the user.
- Develop a framework that allows us to combine provenance data with the outputs from other trust enabling technologies in order to enhance the ability to determine the trustworthiness of information.

To achieve the stated objectives, the project followed three strands, namely scientific development, technical development and applied cognitive psychology. The scientific development includes the study of provenance and trustworthiness of information in order to assess the confidence that a user can have in information. The technical development defines the architecture of a trustworthiness assessment framework and develops a prototype framework based on the architecture. The cognitive psychology strand is to ensure that the right information is presented in the right way to the users. The work presented in this thesis contributes to the first two strands.

## 1.4 Thesis structure

The remaining of the thesis is divided into nine more chapters. The first two chapters (Chapter 2, and 3) are motivational that inspire the development of a Decision Support Framework for crisis management. Chapter 4 is also motivational for the framework in the sense that it suggests a method of prioritising information in the case of uncertainty, especially, when the information comes from untrusted sources. Chapter 5 presents the main contribution of the thesis, the Decision Support Framework, which is elaborated in later chapters (Chapter 6, 7, 8, and 9).

**Chapter 2:** This chapter surveys the relevant literature and presents the state-of-the-art of different aspects underpinning my approach to the automated manage-

ment of crisis information. The aspects of crisis management covered in the survey include:

- Use of open-source data, social networks and crowdsourcing in crisis management (situation awareness, incident detection and early warning, etc.)
- Mutual trust evaluation of people on social networks
- Provenance and trustworthiness of information for crisis management
- Verification of crowd-sourced information
- Consistency evaluation of information for crisis management

**Chapter 3:** Researchers are discussing whether “Social Media” is a “Source of Information or Bunch of Noise” [40]. However, following my own hypothesis that the social media contains both (true) information and misinformation that creates an uncertainty, I have carried out two case studies by collecting data from Twitter, in order to find the fact and assess the utility of the information found in social networks. The case studies include the Mumbai Terrorist Attacks in 2008 and the 2010 Haiti Earthquake. This chapter also elucidates the data harvesting methods and the challenges faced when I attempted to harvest the data from Twitter.

**Chapter 4:** This chapter presents a detailed discussion on the factors that influence people’s trust in information. The factors discussed in this chapter are the most predominant factors that are used in both research and application domains. The factors can be broadly classified into three main categories: factors relating to the provenance of information, factors relating to the quality of information itself and factors relating to the information consumer [103]. The vulnerabilities of these factors to manipulation are also highlighted in this chapter.

**Chapter 5:** This chapter presents and elaborates on the architecture and design of the Decision Support System (DSS) shown in Figure 1.2. The Decision Support System (DSS) helps to evaluate the trustworthiness of information based on some given criteria/preferences and forms one or more possible pictures of a situation (referred to as World Views) that the evidence suggests. The possible heuristics of assigning different scores to different trust/provenance factors is described and how different scores should be assigned to the trust and provenance factors is also exemplified in this chapter. This chapter also investigates different methods of



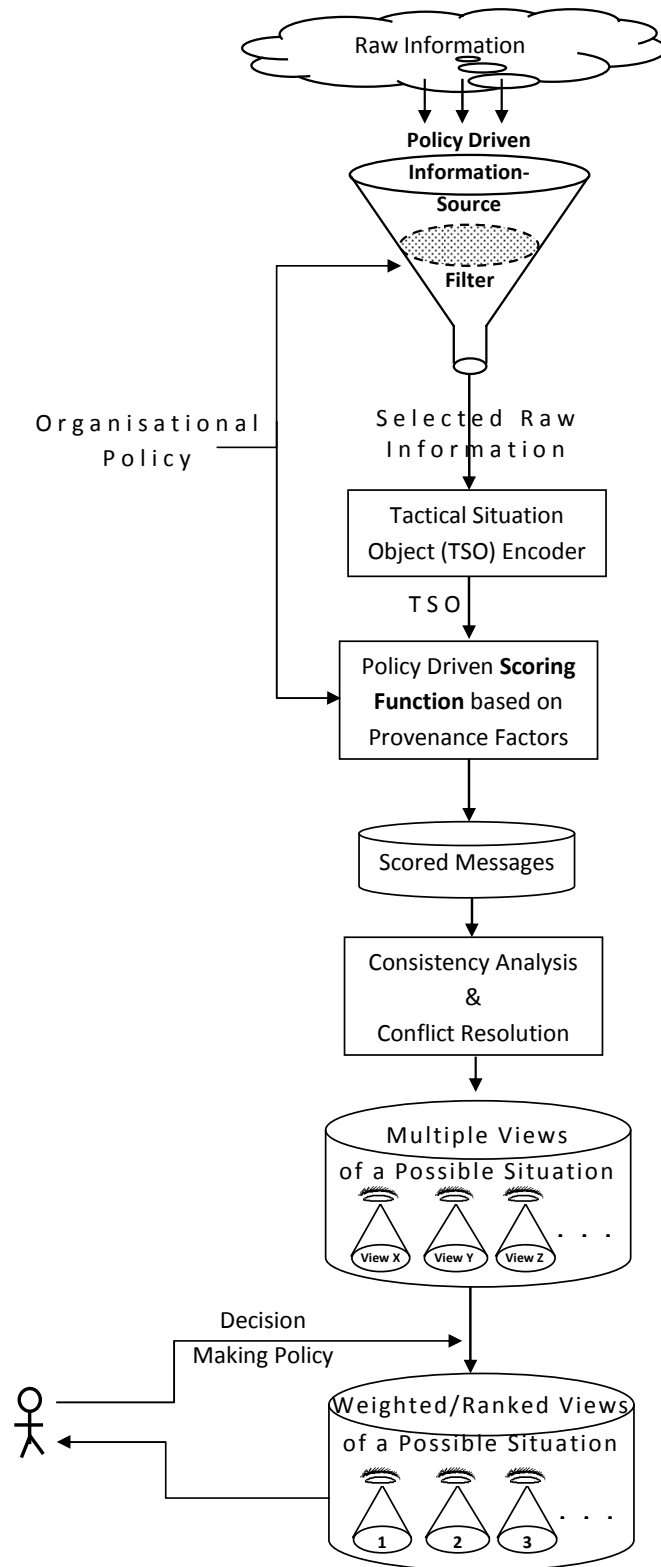


Figure 1.2: Decision Support System Architecture

calculating an aggregate score for each of the world views so that an appropriate overall score is assigned to the world views, which will reflect their true reliability or trustworthiness.

**Chapter 6:** Data from different heterogeneous sources need to be structured into a single format for that will make the data processing and analysis tractable. Although the Twitter APIs return data in XML (and JSON) format, the actual message contained in a tweet is written in a plain natural language. The structure of the tweet is irrelevant to the semantics of the message, i.e. the structure of the tweet is not designed to express its meaning, in any way. Therefore, I have used an intermediate format for encoding the messages that are originally written in a natural language, which facilitates the automated analysis of the messages. The data format which I have taken as a standard is called Tactical Situation Object (TSO). This encoded form of data (TSO) is the input to the scoring function and the (currently manual) Alloy encoding. Tactical Situation Object (TSO) was originally designed for exchanging information between systems during disaster and emergency management [43]. This chapter describes how data can be converted into (the single and uniform structure of) TSO. The structure of TSO and how it has been manipulated are also described in this chapter.

**Chapter 7:** One of the core components of my Decision Support System is the Consistency Analysis and Conflict Resolution Unit. This chapter presents both semantic and predicate logic based approaches to construct world views. Predicate logic based consistency-checking using Alloy has been discussed in detail. This chapter demonstrates that logic based consistency analysis of messages can be performed mechanically. Along with the overall methodology of constructing world views, how different messages can be clustered to generate world views and how the soundness of the world views may be proved have been described in this chapter.

**Chapter 8:** The implementation of the entire decision support system and its usage have been described in this chapter.

This chapter also provides a summary on what programming languages have been used to develop different components of the decision support system, the interfaces between its each component (i.e. what the input and output data and data-formats of each component) and the third party APIs, tools and services (e.g. Analyzer , Google Maps, Google Geocoding Service, and Google Charts.) used to implement the system. How messages can be encoded into TSO relatively easily

using the auto-generated TSO schema has also been demonstrated in this chapter. Other main topics discussed in this chapter include viewing the TSO Data and scoring the messages based on a contrived Organisational Policy, encoding TSO into Alloy and automated analysis of Alloy models using Alloy Analyzer, construction of consistent world views through Alloy automation, presenting the generated world views to the user, and the user interface of the decision support system. I have also provided the metrics on the Lines of Code (LOC) that I have written to develop the Decision Support System (DSS). The Lines of Code metrics will provide an estimation of how much effort may be necessary to reproduce the system.

**Chapter 9:** This chapter provides a step by step demonstration on how a user can use the system with some example data. This includes where (at which stage) the user needs to perform some manual processing and where the processing is automatic. In this demonstration, the user gets a clear understanding of what the input and output of each component of the system are. I have also carried out some experiments in this chapter to demonstrate the effectiveness of the system. The experiment results demonstrate the logical soundness of the world views generated by the DSS. The experiment results also highlight one of the scalability issues in the automatic generation of world views. From the experiment results, this chapter also raises a research question which is discussed in more detail in the following chapter: whether the fully consistent world views or the world views with some fuzziness will be more useful to understand a situation.

**Chapter 10:** This chapter concludes the thesis with the summary of achievements, limitations of the work and the scope of future improvement. This chapter also presents a detailed discussion on whether a fully consistent world view will be more useful to produce a picture of a situation or a slightly fuzzy world view with some ambiguity in it can perform better in this respect. I argue that it is possible to obtain both fuzzy and consistent views of a situation by following the semantic based approach of constructing world views (i.e. possible pictures of a situation), which is discussed in the beginning of Chapter 7. However, I have also provided an alternative way of constructing fuzzy world views by using the concept of fuzzy sets (in Fuzzy Logic) for future development.

## CHAPTER 2

# Managing Crisis Information - State of the Art

Situational awareness is a key requirement for effective decision-making in crisis and emergency situations. Since major incidents, accidents and natural disasters create extremely confusing situations, having correct and reliable information is a prerequisite for gaining intelligence and managing these situations. Hence, it is imperative to collect reliable and relevant information from all possible sources. Traditionally, emergency responders and humanitarian organisations depend, or tend to depend on the official data collected from known reliable sources which is easier to verify [143]. These official sources include trained staff, professional journalists, news outlets, different sensors e.g. CCTV, RADAR, satellite, mobile phone, etc. A significant amount of research has been carried out to improve the situational awareness and emergency response utilising or based on those official information sources [28, 67, 124, 168, 53, 78]. Although a huge amount of real-time information becomes available on the web, especially on the social networks, in the event of most crisis and emergency situations, many humanitarian organisations are still not willing to use information from these sources, as they are deemed unverifiable and unreliable [143]. To find the justification of the emergency responders' lack of confidence on such crowdsourced information and to know the current state of the art, I survey the existing research literature and tools that use open-source data for crisis management.

## 2.1 Data Structure and Sharing Crisis and Situational Information

It has always been difficult for computer systems to process unstructured data. Data needs to be encoded into a suitable machine-readable format before it gets processed by an automated system. This has led to the invention of numerous data formats, such as XML<sup>1</sup> (Atom<sup>2</sup>, RSS<sup>3</sup>), JSON<sup>4</sup>, YAML<sup>5</sup>, etc. However, since different systems are capable of handling different data formats, data interoperability becomes another issue when multiple information systems need to work together. This data interoperability issue becomes even worse, when it comes to dealing with any large-scale disaster and emergency situation which often requires different emergency services, sometimes from different countries, to operate together. This causes a serious coordination problem, especially in a disaster scenario. To overcome this problem, it is essential to establish a standard data exchange format, especially for the emergency responders, for sharing situational information in the context of crisis and emergency management. OASIS [42], a European Framework 6 project, took an initiative to develop a message structure for the exchange of disaster and crisis information between computer-based systems in such a way that it can be reliably coded and decoded. Their specified message structure for sharing situational information, referred to as Tactical Situation Object (TSO), was formally accepted by CEN Workshop Agreement (CWA) [43]. TSO comes with a rich data dictionary that provides all codes that can be used in a TSO.

## 2.2 Use of Open-Source Data, Social Networks and Crowdsourcing in Crisis Management

A useful feature of social networks is that people tend to respond to certain events and incidents in real time. For example, when a disaster happens, natural (e.g. earthquake) or man-made (e.g. accidental or terrorist incident), social networks receive a torrent of message as people start sharing information about the incident.

---

<sup>1</sup><http://www.w3.org/XML/>

<sup>2</sup><http://tools.ietf.org/html/rfc4287/>

<sup>3</sup><http://validator.w3.org/feed/docs/rss2.html>

<sup>4</sup><http://tools.ietf.org/html/rfc7159>

<sup>5</sup>[www.yaml.org/spec/1.2/spec.html](http://www.yaml.org/spec/1.2/spec.html)

Numerous studies have been carried out either for understanding the utility of open-source information in case of crisis and emergency, or for utilising such information from the web and social networks for crisis management [87, 156, 118, 38, 79, 88, 81, 1]. Effort has also been made to collect terrorist information from open-sources on the web in order to analyse, visualise and destabilise terrorist activities/networks [86].

### 2.2.1 Incident Detection and Early Warning

The possibility of using social networks data for detecting incidents and generating early warnings is being investigated by researchers. One such effort has been explained by [116]. This paper utilises the real-time nature of the social networks and explains how an instant detection of incidents may be possible simply by monitoring the current data stream on social networks such as Facebook, Twitter, etc. This may enable the generation of early warnings for some people. This paper presents a system that can distinguish between Twitter messages related to current and past events. Thus, the system can detect earthquakes in certain places where there are a large number of Twitter users by scanning Twitter messages in real-time. According to this paper, earthquakes propagate at about 3-7 km/sec and therefore, the system can generate a relatively early warning message for some of its registered users who are relatively far away from the already affected areas. They use Support Vector Machine (SVM), a machine-learning algorithm, to automatically identify tweets that refer to a target event (earthquake). Bayesian filters, specifically Kalman filters and particle filters, are used for estimating the centres of earthquakes. They also use a probabilistic model to minimize the number of false alarms of earthquakes. However, this work does not provide any further information about the aftermath of the disaster.

There are other systems that use social media data to detect incidents or extract information related to incidents [68, 164]. For example, a semi-automatic system has been proposed in [68] with a view to detect emerging events/incidents, after harvesting information from open-sources using a third-party tool, Recorded Future<sup>6</sup>. Recorded Future uses Named Entity Recognition or Entity Extraction techniques, to extract different event information. Named Entity Recognition is an information extraction process that recognises and labels sequences of words extracted from a piece of text which are the names of things (e.g. person, location

---

<sup>6</sup><https://www.recordedfuture.com/this-is-recorded-future/>

or place, organisation, etc.), dates, numbers including telephone numbers, and so on [51]. For example, Recorded Future can detect an earthquake as the Named Entity Recogniser is able to extract the highlighted information from the following message: “*Earthquake hit Haiti at 4:53pm (local time) on 12 January 2010*”. The proposed system is also dependent on Ushahidi<sup>7</sup> platform for collecting information from the people on the ground, who submit messages directly to Ushahidi platform by sending SMS, email, etc. Information collected using both Recorded Future and Ushahidi is then combined manually. Information related to different events is displayed on the map using Ushahidi mapping tools using the available geographic locations of the events.

Although it is possible to detect incidents to generate (relatively) early warnings for some people in some places, these systems (mentioned above) may generate false alarms too. If a crime syndicate can manage to generate a large volume of false messages on the social networks, then the above mentioned systems are likely to give false alarms as they do not have any mechanism to evaluate the reliability of the information sources or the veracity of the information.

### 2.2.2 Situation Awareness

With the realisation of the fact that the information freely available on social networks can help improve situation awareness in the cases of crisis and/or emergency, efforts are being made (as witnessed in [19] and [164]) to utilise the opportunity. The system described in “ESA: Emergency Situation Awareness via Microbloggers” [164] not only can detect incidents by analysing user activities on Twitter but also collects relevant information from there to improve situational awareness. This ESA system works in the following steps:

**Burst Detection:** This system continuously monitors the Twitter stream and detects sudden bursts of tweets in order to detect an incident. It exploits the statistical incidence of words used on Twitter to describe emergency events. It uses a time-based probabilistic method [47] to identify the predominant words (“bursty words”) from a list of tweets. The ESA system trains itself with historical data collected before, during and after past incidents in order to assess the probability of occurrence for a term/word and build a “language model”. An alert is made indicating a potential incident when the probability distribution of a word/term significantly deviates from the language model.

---

<sup>7</sup>[www.usahidi.com](http://www.usahidi.com)

**Tweet Clustering:** The system clusters the related tweets (even though they may be contradictory messages) by using third party tools such as Apache Solr<sup>8</sup>, an open-source search engine, and the Carrot2 clustering engine<sup>9</sup>. It filters the tweets that contain useful information about a particular incident by building statistical classifiers. A variety of features extracted from tweets are used in the classification process.

**Geo-Tagging and Incident Augmentation:** When the location information is available for a tweet, ESA tags the tweets with its geocodes and displays the content of a tweet at the specified location on a map. If a tweet is not already geotagged by its author, the ESA uses the registered location from the user profile. ESA also extracts the entities (e.g. names of people, organisations, locations, date and time, etc.) that are mentioned in tweets using the Stanford Named Entity Recogniser (NER)<sup>10</sup>. Extracted locations and organisations are then marked on the map to show the incident locations.

The main strength of this system is Twitter burst detection and tweet clustering. However, it does not give a clear view of the situation by checking the consistency of the messages and separating the contradictory messages into different ‘world views’. In fact, ESA cannot identify whether the messages are supporting or contradicting each other; it only clusters the related messages.

An on-going work to detect, assess, summarise, and disseminate information related to a target incident has been presented in [19]. The techniques presented in [19] are very similar to those used in [164].

The use of social networks, more specifically Twitter, has also been studied in the context of a terrorist attack in [105]. While the crisis related information shared on social networks is useful for the general public, main stream media and government agencies, recorded conversation between one of the on-site terrorists (say, ‘Terrorist (on-site)’) and one of their accomplices (say, ‘Terrorist (off-site)’) from a remote location, during the Mumbai attacks, shows that such situational information was followed and used by the terrorists too [71](taken from [105]):

**Terrorist (off-site):** *“See, the media is saying that you guys are now in room no. 360 or 361. How did they come to know the room you guys are in?...Is there*

---

<sup>8</sup><https://lucene.apache.org/solr/>

<sup>9</sup><http://project.carrot2.org/>

<sup>10</sup><http://nlp.stanford.edu/software/CRF-NER.shtml>



*a camera installed there? Switch off all the lights...If you spot a camera, fire on it...see, they should not know at any cost how many of you are in the hotel, what condition you are in, where you are, things like that ...”*

**Terrorist (on-site):** *“I don’t know how it happened...I can’t see a camera anywhere”*

Sometimes, the information shared on social networks contains sensitive and strategic information related to security missions. Therefore, an information control framework has been proposed in [105], so that malicious parties cannot access the sensitive and critical real-time information.

Researchers have also made efforts to investigate how semantic web related technologies can be used to improve situational awareness in the context of humanitarian operations [132]. They propose to develop a semantically rich data repository of real-world events with the information collected from disparate and semantically heterogeneous sources, and provide a reasoning service to fuse information, considering the level of trust and confidence assigned to the information sources. However, they have ignored the fact that there may be some contradictory information. Hence, their system does not provide any mechanism for dealing with the contradictory information. Description logic based reasoning service of their system is used only to fuse information from disparate sources, not for finding or resolving contradictions between messages.

### 2.2.3 Trust Evaluation

A quantitative measure of trust has been introduced in [2] by analysing people’s behaviour and activities on social networks. The underlying notion that has been used to measure trust is that when two people trust each other, they express this trust through their behaviour. One such behaviour is that they are likely to converse with each other frequently. The authors focus on two particular behaviours as an expression of trust: conversation and propagation. The authors claim that trust among people can be measured by quantitative analysis of their conversation. People’s trust measured from their conversation is referred to as ‘conversational trust’. The measure of conversational trust is based on the following assumptions:

- Longer conversations imply more trust.

- Frequent conversations imply more trust.
- Balanced participation by A and B implies more trust.

They define the conversational trust  $Tc(A, B)$  as follows:

$$Tc(A, B) = \sum_{i=1}^l ||C_i|| \cdot H(C_i)$$

where,

- $C_i$  is a conversation between A and B containing two or more messages that were exchanged in close proximity of time.
- $H(C_i)$  is a measure of the balance in the conversation.

They use the Binary Entropy Function [ $H(p) = -p \log p - (1 - p) \log(1 - p)$ ] to measure balance i.e.

$$H(C_i) = -p(C_i) \log p(C_i) - (1 - p(C_i)) \log(1 - p(C_i))$$

where,  $p(C_i)$  is the fraction of messages in the conversation  $C_i$  that were sent by A. The entropy function is used to measure balance because a conversation is said to be ‘100% balanced’ when half of the total messages in the conversation is sent by A and the rest half is sent by B i.e.  $H(C_i)$  is maximum when  $p(C_i) = \frac{1}{2}$

In social networks, people propagate messages received from others (as they ‘Share’ and ‘Retweet’). The authors have proposed another method of measuring trust which depends on the propagation of messages. If a person ‘X’ regularly forwards the message that s/he receives from person ‘Y’, then it can be said that ‘X’ trusts ‘Y’. Unlike conversational trust, ‘propagational trust’ is directed. B may propagate information received from A but A may not (propagate information received from B). They propose two different formulae to measure the directed trust weight  $Tp(B, A)$  from B to A:

$$(i) \quad Tp(B, A) = \frac{prop_{AB}}{prop_B} \qquad (ii) \quad Tp(B, A) = \frac{prop_{AB}}{m_{AB}}$$

where,

- $m_{AB}$  is the total number of messages that B received from A,
- $prop_B$  is the total number of messages that B received and subsequently propagated

- $prop_{AB}$  is the number of messages that  $B$  (received from  $A$  alone) and propagated them

The first formula indicates, how much  $B$  trusts  $A$  as compared to others who appear to be trustworthy to him ( $B$ ), based on his propagation statistics, assuming that propagation is an indication of Trust. The second formula indicates what fraction of the messages sent by  $A$  to  $B$  is considered as credible by  $B$ , based on the notion that forwarding a message means believing that message.

Although the authors have carried out experimentation with Twitter data and the experiment result has supported their hypotheses, they acknowledge that people's trust is context dependent as people's trustworthiness is context dependent in the first place [17, 123, 15]. Forwarding of messages does not necessarily indicate believing/trusting the messages or message senders. People may share/forward information which they find interesting, strange and unusual or amusing. However, even if this method is useful to measure trust between social network users (information sources/agents), it is unlikely to be useful for others to assess the reliability of a piece of information received from either of those users. This may, however, influence our judgement about the credibility of a message (a piece of information) if we know that either or both of  $A$  and  $B$  are very trustworthy, they are very competent in judging credibility of information and they share information only if that information is credible to them. Measuring trust based on the frequency and length of conversation is likely to give wrong information as friends or social/business partners, who do not trust each other any more, may be found involved in frequent and long arguments. It will be a fallacy if we think that they trust each other because of their frequent and lengthy conversation or communication. However, it may be a valid assumption that the subject of their conversation is something that both of them are interested in or concerned about.

### 2.3 Provenance and Trustworthiness of Information for Crisis Management

One of the biggest problems of using open and crowd-sourced information is that it contains a large amount of noise and unreliable information including hoaxes [111]. If a system is given garbage as input then undoubtedly, its output will be garbage too, regardless how good the system is. Therefore, it is imperative to assess the

quality and reliability of information before it is fed into a system. When it comes to dealing with any crisis or emergency situation, the importance of using reliable and trustworthy information is even higher. Therefore, the humanitarian organisations do not want to use information from unknown and unreliable sources as they are deemed unverifiable and untrustworthy [143]. As a result, credibility study of open and crowd-sourced information for crisis management is now an active research topic [111, 20, 90, 87]. Researchers have studied a large volume of tweets relating to 2010 earthquake in Chile in order to assess the reliability of Twitter as an information source during crisis or emergency situation [87]. During major incidents, it has been found that comparatively a limited vocabulary is used on social networks e.g. Twitter. This indicates that people mostly discuss about a common topic during a crisis or emergency situation. Researchers have also found that false information is questioned much more than confirmed or true information on Twitter [87]. This provides some clue for the users to decide how much to trust a piece of information. This may also make it possible to detect rumours or misinformation automatically through the analysis of tweets.

The quality and trustworthiness of information is often estimated based on its provenance and the trustworthiness of its source or provider [30, 146, 120, 150, 154, 155, 165]. The credibility of the source of crisis information shared on Twitter has been studied in [146]. For doing the analysis, the authors categorise the messages based on user, location, language, type, and credibility of the source. While it finds that the vast majority of the information shared on Twitter are actually taken from others (not from first-hand experience), 70% of such information was taken from highly credible sources. However, this work has the following limitations:

1. Poor credibility assessment method: They assess the credibility of a source based on the source's perceived ability and intention to provide correct information. Since the perceived value of something is highly subjective, perceived ability and intention of a data source cannot be a good indication of its credibility.

The authors also consider credibility as a binary factor that can have either of two values: high or low.

2. No attempt has been made for any logical evaluation of the messages (i.e. the claims made in the messages). This means that if a perceived credible source says that the sun rises in the west, then this will be accepted as true.

3. Registered locations found in Twitter user profiles are used as the (true) location of the users, whereas the authors acknowledge the fact that at least 34% Twitter users in the US do not provide real or meaningful location information in their profiles [60].

A data provenance trust model has been proposed in [30] for estimating the level of trustworthiness of both information and its sources. This trust model mainly considers the ‘reputation of the source’, ‘corroboration’ and ‘contradiction of messages’ to evaluate the trustworthiness of both data and data providers. To verify corroboration and to detect collusion, the notions of ‘data similarity’ and ‘path similarity’ have been introduced. While the data similarity helps to detect possible corroboration, ‘path similarity’ helps to detect collusion by comparing the propagation paths of same/similar messages. Based on these factors, the trust model assigns a trust score which may enable the data users to decide whether to use the data or not. However, the method of assessing the trustworthiness of a source is too simple. Although the authors of [30] acknowledge that erroneous data may be generated by sources having inadequate knowledge or malicious intent, they assess the trustworthiness of a source based on its reputation only i.e. the amount of correct/incorrect information it has provided before. Since, there is no guarantee that a data source with high reputation cannot be malicious and reputation is the only factor that has been used to assess the trustworthiness of a source (information provider), this trust model is likely to fall short of desired performance. Another limitation of this work is that it does not use any information that has been reported by a single source (i.e. it has not been corroborated by others), regardless of how trustworthy the source is. This approach may lead to a grievous consequence especially in the event of crisis and emergency should important information be overlooked or ignored.

Another provenance based trust model has been presented for multi-hop networks in [154] which uses similar concepts found in [30] but does not consider the possibility of collusion attacks.

Authors of [89] also shed light on the evaluation of trustworthiness of real-time information. They suggest that when people believe that a piece of information is trustworthy, they do so for “intrinsic and/or extrinsic reasons”. The ‘intrinsic’ reason is that the given information matches with people’s prior knowledge or belief. When people believe a piece of information because they trust its information source(s), the authors refer to this as ‘extrinsic reason’. According to the authors of [89], when people do not find any of these two reasons, intrinsic or extrinsic,

they look out for corroboration from independent sources. Thus, the overall trustworthiness of information depends on the credibility of the information itself, the reputation/reliability of its original and intermediate sources, and independent corroboration. It should be noted that data provenance has been widely investigated in various other fields/contexts, in addition to crisis management, such as scientific workflows, database, museum work, multi-hop networks, etc. [154]. Some also consider provenance as a security control in its own right, because of its direct link to some security factors such as ‘information integrity’ and ‘non-repudiation’ [80]. In some cases, provenance can also protect ‘confidentiality’ and ‘access control’ [80].

## 2.4 Verification of Crowd-sourced Information

After realising the value of crowd-sourced information, the main concern now is being able to successfully utilise the information. Verification of information becomes essential at times, in order to ensure successful use of such information. I have surveyed some of the cases wherein verification of crowd-sourced information has been performed.

### 2.4.1 DARPA Network Challenge

In December 2009, Defense Advanced Research Projects Agency (DARPA) organised a competition by placing ten red weather balloons at undisclosed locations across the United States [3]. A team of researchers from the Massachusetts Institute of Technology’s Media Lab won the \$40,000 prize money, by being the first to locate all of the balloons [112]. They located all ten balloons in less than nine hours by crowdsourcing with the help of social-networking technologies. There were mainly two tasks in solving the problem. Firstly, networking and mobilising people. In fact, MIT Media Lab Team recruited about 5,400 individuals in approximately 36 hours [4]. The winning strategy of the MIT team for this task was to use the prize money as a financial incentive for not only the people who correctly located balloons but also for those who affiliated the balloon finder with the MIT team. They allocated \$4,000 for each correctly located balloon. Distributing the prize money to everyone involved in successful location of each balloon motivated more than 5,000 people to join the team, including some from outside of the U.S [141]. Because of the broad reward scheme, people from outside U.S. could also be rewarded simply for knowing someone who could find a balloon. The second task involved identifying

the accurate balloon locations from the incorrect locations and this was the hardest part of the challenge [4]. Not surprisingly, people submitted more incorrect locations than correct locations. According to one report, 124 reported locations were incorrect out of 186 reports that MIT received [94]. According to another report, 30 to 40 reports on balloon sighting were accurate out of more than 200 submissions that the MIT team received [141]. However, for verifying information, the MIT team mainly relied on human intelligence for manual analysis and reasoning about the information on balloon sightings, and to eliminate information with inconsistencies. The first strategy was to observe the differences and similarity of the location coordinates. Since all of the balloons were located in open public spaces, multiple submissions of coordinates for each balloon were expected. Hence, corroboration was given more importance in selecting a coordinate as a correct location. However, their adversaries also made multiple submissions of the same fake coordinates. The members of the MIT team noticed that each of the genuine coordinates reported for the same balloon naturally differed slightly from each other, whereas the fake reports contained identical coordinates for a specific balloon which made them doubtful [141]. Another strategy that the MIT team used was checking reporter's IP address in order to get an idea about the location of the reporter and to check whether it matches with the reported location of the balloon. In one occasion, for example, a false report of a balloon sighting in Florida came from an IP address in the Los Angeles area [141]. The GTRI team (also known as "I Spy a Red Balloon") that successfully located 9 balloons used a very manual method of verifying information. They directly contacted local people who live or work in the vicinity of claimed balloon locations.

Although, the red balloon challenge demonstrated that some otherwise difficult tasks may be accomplished relatively quickly and easily by crowdsourcing the task, the verification and consistency analysis of information was performed almost entirely manually using human intelligence.

#### 2.4.2 Referral-based Verification

A model has been proposed in [94] for verifying information in a special crowdsourcing setting referred to as 'referral-based crowdsourcing' where each report is passed from one person to another through a referral chain. The model assumes that there is a financial or other incentive for the reporters who report true/correct information and everyone except an "irrational" person will be motivated by the

incentive i.e. reporters should not have any other motivation (or utility) other than receiving the offered incentive. Depending on this notion that no one except an irrational person will ever want to lose the incentive, the model includes a reward or penalty scheme for providing correct or incorrect information. The model also assumes that information can be verified with certainty using crowdsourcing. Hence, the model specifies the minimum reward and penalty necessary to encourage people to verify the reports before confirming as true. The model also presumes that wrong information will be given only by mistakes, not with a malicious intent.

The model is only applicable in certain scenarios where there is really a tangible reward for providing correct information; whereas in most situations, especially in case of natural disasters, altruism is one of the main incentives for people to provide true and correct information. Even if there is a tangible reward for providing true information, yet this model is ineffective in certain cases as the authors acknowledge that “irrational agents are not affected by penalties and compensation: they lie irrespective of the incentives” [94].

### 2.4.3 Ushahidi - Swift River

Swift River is a platform that facilitates the process of data validation. Ushahidi<sup>11</sup>, a platform for mapping crisis information, uses Swift River for validating crowd-sourced information [113]. Although there is some automation in Swift River such as filtering/flagging duplicate information, its validation process is entirely manual. It uses crowdsourcing (direct involvement of human intelligence) to validate crowd-sourced information in near real time. Swift River requires trained people to assign veracity scores to the information gathered through crowdsourcing (which indicates the likelihood that the said events have occurred) [134]. A serious limitation of Swift River is that it depends on human intelligence at three levels:

1. Collect information from untrusted sources, which they refer to as Unbounded Crowdsourcing
2. Verify information with the help of their (trusted?) field workers/volunteers, which they refer to as Bounded Crowdsourcing [85].
3. Assign a veracity score to the collected information with the help of trained staff.

---

<sup>11</sup>[www.ushahidi.com](http://www.ushahidi.com)



## 2.5 Consistency Evaluation of Information for Crisis Management

Reasoning and consistency checking is generally a well-studied topic [21, 61, 129, 9, 35, 152, 48, 14, 37, 137]. Automated reasoning and consistency checking have been used in various application domains including verification of software and hardware specification [106, 137], fault diagnosis [37], semantic web ontology [9, 36, 35, 152], UML Models [5, 128], and medical diagnosis. Some of the prominent formal reasoning methods are theorem proving, model checking, model finding, constraint solving, etc. These methods are automated using various tools that are based on propositional logic and SAT (Boolean Satisfiability problem) solvers [129, 130], first-order and higher-order logic [106], logic programming such as Prolog [9], graph and set theory [137], description logic [5, 128, 132] and ontology based reasoning [121, 131]. Each of the reasoning methods and tools has strengths and limitations. For example, it is comparatively much easier to automate the propositional (Boolean) logic based reasoning methods as the SAT (Boolean Satisfiability) problems are decidable [66] (i.e. we can always construct the truth table for a propositional formula). This is why SAT solvers are extensively used in automated reasoning and consistency checking. However, a fundamental problem of propositional logic is that the expressibility (expressing power) of propositional logic is very limited as it does not support any quantifier. In fact, the more expressive a formalism is, the harder it is to prove; consequently, the harder it is to automate [55]. This is where the limitation of first order and higher order logics stems from; they are more expressive and therefore, their automation is harder. However, first order logic is a better choice for automated reasoning because of its moderate expressibility (although less expressive than higher logics) and the availability of mechanical (automatic) tools for automation.

Logic-programming languages such as Prolog provide another option for the automation of logical reasoning and consistency checking. However, Prolog's reasoning capacity is known to be limited [9]. Although Prolog is a logic programming language which works using unification and resolution techniques [157], it is logically imperfect and does not offer the full power of resolution theorem proving [119, 160].

Some of the reasoning methods are based on graph and set theory [137]. However, graph theory based knowledge representation and reasoning is very primitive and less expressive as they use elementary notions of set theory that have graphical

representations (e.g. sets, elements and relations) [137].

Model checking algorithms based on first order predicate logic are popular in automated reasoning and verification [106]. However, model checkers usually suffer from state explosion and thereby, generally unsuitable for checking systems with finite but large state space [29, 106, 23, 162, 163], although recent experiments demonstrate the successful use of ProB<sup>12</sup>, an animator and model checker for the B-Method<sup>13</sup>, for the validation of large railways data [76]. Model checking algorithms explore the entire state space (the whole set of possible interpretations of variables) to ensure that a formula holds in all states.

The reasoning method which does not suffer from state explosion is theorem proving which is, therefore, suitable for reasoning with systems having large and complex data structure [59]. However, theorem proving is not suitable for using domain specific knowledge [16, 167]. Therefore, the reasoning performed by theorem provers is based on syntax and not semantics. Thus, the use of theorem proving techniques becomes unrealistic in reasoning with a knowledge base containing situational information. Another drawback of theorem provers is that they often require a great deal of user expertise, effort and assistance [59, 6] ([6] is taken from [106]). Therefore, theorem provers are generally not automatic as their automation is difficult and costly [59].

There is a plethora of research articles that have studied crowd-sourced information in the context of crisis and emergency [60, 87, 146, 118, 38, 116]. However, I have not found any related work which uses formal logic (by means of formal modelling tools or semantic reasoners) for checking consistency of information to resolve contradiction between different pieces of information from untrusted sources, in any context including crisis and/or emergency. However, I have found a few attempts made by the researchers to exploit the reasoning facility provided by ontology languages e.g. OWL (Web Ontology Language), in order to perform inference and answering questions [121, 131], not for finding or separating contradictory information.

Although semantic web related languages and tools provide some reasoning capability, their reasoning power is still primitive [152]. Therefore, researchers advocate for using formal modelling tools, such as Alloy, to provide an automated reasoning service for the semantic web ontology languages too [152]. There are also examples of using Alloy for analysing and checking the consistency of semantic web ontol-

---

<sup>12</sup><http://www.stups.uni-duesseldorf.de/ProB/>

<sup>13</sup><http://www.methode-b.com/en/>

ogy [35, 36].

Some researchers have used statistical methods, more specifically Dempster Shafer Theory, for reasoning under uncertainty in the context of crisis and/or emergency [41]. It is true that the Dempster Shafer Theory (DST), as well as Bayesian Networks (BN), is a good method of reasoning with uncertainty, especially in diagnosing problems (medical or other). Here is a basic but a typical example application scenario of these methods:

If a patient has *fever, dry cough, headache, runny or blocked nose, tiredness, chills and aching muscles*, then a reasoning system which uses a statistical method (DST or BN) may infer from the symptoms that the patient has caught influenza.

It is important to note that the Dempster Shafer Theory or Bayesian Network infers something new that was unknown (*influenza*, in our example) from the existing knowledge (symptoms, in our example). The reason why it is not appropriate to use any of the statistical reasoning methods (in this research) is that the aim of this research is to deduce the fact(s) from the information in hand without creating any new information. For example, after knowing the symptoms of the patient, if someone says that the patient has caught malaria while someone else says the patient has caught influenza, then I want my system to say that the patient is likely to be suffering from influenza or malaria. The system may also give more priority to one possibility (e.g. influenza) over the other, depending on various provenance and quality factors that influence trust in information such as reputation, competence, etc. I do not want to add any new information to increase the information overload for the end user which will ruin the purpose of developing the system. I leave the inductive reasoning with the end user. However, it is one of my future plans to see whether the statistical methods or Fuzzy logic methods may be used to improve the result.

## 2.6 Summary

Although the study of social networks has a long history [91] and social networks have now emerged as a big and potentially useful data source, there is no simple and automated way to assess the reliability of the information available on social networks. Although the method shown in [2] to compute ‘conversational trust’ and ‘propagational trust’ may be useful to measure trust between two agents (in-

formation sources), it is unlikely to be useful to assess the reliability of a piece of crowd-sourced information (unless one of the agents/sources are very trusted to us). Therefore, it is still difficult to use such (unreliable) information with confidence in any serious case, and emergency and/or humanitarian responders are sometimes unwilling to use this information [143]. This highlights the fact that more effort needs to be made to utilise the information for crisis management. However, it is not possible to verify a piece of (crowd-sourced) information without having an access to the ‘real world’ and the literature survey (in Section 2.4) shows that the verification process is almost entirely manual. Hence, my approach to achieving my second research objective is to construct the possible views of a situation (world view), which the evidence pertains to, and I will not verify whether the information is true or false. Verification of information and finding the truth may be a future research topic.

Information related to a target event may be filtered and/or clustered using techniques described in [116], which involves Support Vector Machine (SVM), and in [164], which involves Apache Solr<sup>14</sup> and the Carrot2 clustering engine<sup>15</sup>.

Automated reasoning can help in establishing the reliability of information which is consistent with our knowledge. However, there is no such formal reasoning method or tool which is equally applicable and suitable in all contexts. Each of them has strengths and weaknesses. Since automated reasoning is one of the requirements of my decision support system, after considering the pros and cons of various reasoning tools and techniques, model finding based on domain specific knowledge appears to be the best choice for reasoning with situational information. In my research context, model finding tools are more ideal than model checkers as I do not need to check the entire state space when a satisfying instance is found. Hence, Alloy, which combines the power of first order logic, set theory, and SAT solvers, appears to be an appropriate tool for automated reasoning and consistency checking in this context. More about Alloy and Alloy Analyzer will be discussed in Section 7.2.

---

<sup>14</sup>An open-source search engine, <https://lucene.apache.org/solr/>

<sup>15</sup><http://project.carrot2.org/>

## CHAPTER 3

# Uncertainty in the Wake of Crisis and Emergency

Social networks (e.g. Facebook, Twitter) and Photo/Video hosting services (e.g. Flickr) received a large amount of postings during a series of coordinated attacks that took place in different locations of Mumbai shortly before 10:00 pm on 26 November 2008 [12, 159]. Twitter started to receive messages from eye witnesses and others, both from inside and outside the affected locations, immediately after the incident [24]. During the attacks, eyewitnesses sent an estimated 80 SMS to Twitter every 5 seconds i.e. about 1000 SMS messages per minute [24]. Twitter and Facebook were also flooded with updates soon after the January 2010 Haiti earthquake of magnitude 7.0. People used these social networks to collect and share information about the disaster and its victims. Twitter posts appeared within seconds of the earthquake [95] and Facebook claims that they were receiving about 1,500 Haiti related messages per minute [95, 26]. Many such social network postings might be useful to agencies charged with responding to such situations (whether natural/accidental disasters or criminal incidents).

The mash-up approach may be used that brings together information from multiple public and specialist sources including social networks to form a synoptic view of a situation. However, in order for such data to have utility, it must be of appropriate quality. While researchers are discussing the topic “Social Media: Source of Information or Bunch of Noise” [40], based on what I have gathered from the literature (particularly [19], [116], [164], [3], and [141]) survey, I hypothesise that social media contains both (true) information and misinformation that creates an uncertainty. It is also likely that the uncertainty lasts only for a specific period (window) of time, which I call ‘Window of Uncertainty’, as the “truth prevails”

eventually.

In order to understand the nature and assess the utility (usefulness) of the information found in social networks, I have carried out two case studies collecting data from Twitter. I elucidate the data harvesting method and the challenges faced when attempting to crowd-source from the social network before presenting the results of the case studies.

### 3.1 Harvesting Open-source Data from Twitter

Twitter, a micro-blogging service which produces 500 million tweets each day [145], receives large volume of messages during crisis and emergency. Along with other features of Twitter, the fixed length (144 characters) of Twitter messages (tweets) makes its use for data analysis more tractable. Therefore, I have decided to use Twitter data for the case studies.

#### 3.1.1 Collecting Tweets from Targeted User Accounts

Collecting tweets from the Twitter website is a very naive approach. The HTML (HyperText Markup Language) pages from the Twitter website need to be scraped for collecting tweets this way. Using the Twitter's search tool and saving the search results manually is not a practical method either. Therefore, I have developed web applications (in C# using the ASP.NET framework) for downloading tweets. One of these applications can retrieve all public tweets that were created between two given dates from any specified user account on Twitter. The Twitter API (Application Programming Interface) used in this application, called *Twitter User Timeline API*, returns data (tweets) in XML (Extensible Markup Language) format. One of the tweets retrieved by this API from a user's timeline is shown in its original structure and entirety in Appendix A.1. However, some of the commonly used elements of tweets is shown below for the purpose of illustration.

```
<status>
  <created_at>Thu Nov 27 06:43:47 +0000 2008</created_at>
  <id>1025980128</id>
  <text>
    #mumbai Times Now quotes British high commission saying 7 British citizens injured in attacks.
    Israelis being held in & near Chabad House
  </text>
  <user>
    <id>11778</id>
```

```
<name>Dina Mehta</name>
<screen_name>dina</screen_name>
<location>Mumbai, India</location>
<time_zone>Mumbai</time_zone>
</user>
<geo />
<coordinates />
</status>
```

After downloading the tweets, the application parses the XML data and saves the tweets along with other related information (metadata) in an SQL Server database<sup>1</sup> for further analysis. It should be noted that although the Twitter API returns data in XML format, the actual message contained in a tweet is written in a natural language. The structure of the tweet is irrelevant to the semantics of the message, i.e. the structure of the tweet is not designed to express its meaning, in any way.

This application is particularly useful to collect information about past incidents. Since, Twitter generates an extremely large volume of data (500 million tweets per day [145]), it is very difficult to collect previous data using the Twitter Search API because of its limitations and the restrictions on its usage. Since, the Search API gives access to an index of recent tweets (not a complete index of all tweets), it is not possible to retrieve those tweets that are approximately a week (6 to 9 days) old by using the Search API [149]. However, the Twitter ‘user\_timeline’ API is not free from restriction either. For example, this API can only return up to 3,200 of a user’s most recent tweets [148]. Nevertheless, the restriction on retrieving individual user’s public tweets is more flexible than that for the Search API. Therefore, if some users are found very active on Twitter and they are disseminating a large amount of valuable information about an incident, then it is easier to collect messages from that user’s Twitter timeline rather than using the search API as the search API will not have access to that information.

---

<sup>1</sup>**Why SQL Server database?** Although, XML is a popular data interchange format that makes exchange of data between different applications on different platforms very easy, it has other limitations too. Since XML files contain textual representation of data and the language (XML) is very verbose, XML documents are larger than binary representations of the same data [84]. Therefore, processing large XML documents can consume too much memory and processing capacity. Secondly, while the Twitter data retrieved from targeted individual accounts comes in XML format, the Twitter Search API (explained in following section) returns data in a different format (JSON). Hence, for further processing and querying the harvested data, I have chosen to convert them (from XML and JSON) into a single format and create a single data-repository by storing the data into an SQL Server database.

### 3.1.2 Data Collection using Twitter Search API

The Twitter Search API facilitates searching with the given keywords. The Search API is useful for retrieving real-time or near real-time information as it searches against the real-time index of recent tweets. The Twitter Search API also provides a few filtering mechanisms that allow us to retrieve tweets made by users from a specific location and in a specific period of time. Therefore, I have developed another web application, by using the Twitter Search API (version 1) [147], in a bid to have a real-time data harvesting tool. Unlike the ‘user.timeline’ API, the search API returns the matching tweets in JSON (JavaScript Object Notation) format, thereby requiring my application to have a JSON parser for extracting and storing the tweets in SQL Server database for later use. However, it is not possible to use the Twitter Search API to find tweets older than about a week and there is also a limit on how frequently a client application can call the API [149].

## 3.2 Reproducing a Graphical Timeline of Incident-related Tweets

For the ease of analysis, to get an insight about an incident from the collected tweets, and to see the pattern of message stream, I have created an application that displays a custom timeline (as shown in Figure 3.1) of who said what about the incident and when. If the database contains information about multiple incidents, then

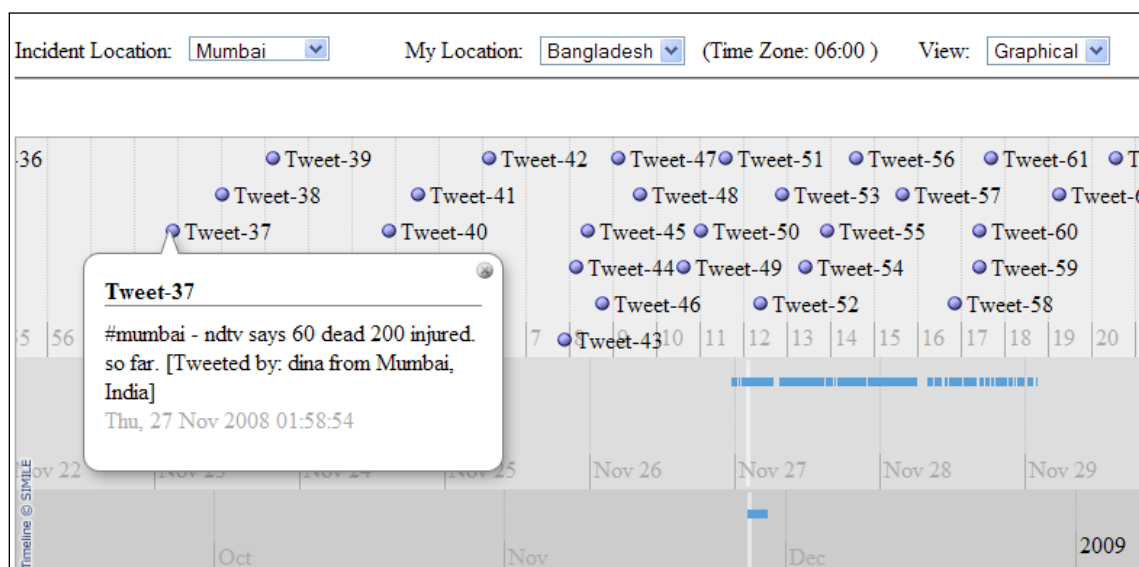


Figure 3.1: Message Timeline

the application shows the timeline for the chosen incident. While the application



can show the timelines both in graphical and tabular view, the timelines can also be viewed in GMT or in a selected local time. I have used a free service provided by MIT<sup>2</sup> for producing these graphical timelines. Since, this timeline service requires the input data in a separate XML file and the structure of the input data is completely different from that of the original Twitter data (although, both are in XML format), I have written another small application that programmatically creates the input XML file by using the Twitter posts saved in the SQL Server database.

### 3.3 Uncertainty in the Crowds during Mumbai Incident 2008

On 26 November 2008, a series of coordinated shooting and bombing attacks took place in different locations of Mumbai which started shortly before 10:00 pm [12, 159]. The siege of different attack locations lasted for about 60 hours before the Indian security forces overpowered the attackers on 29 November, 2008 [11]. During the attacks, eyewitnesses sent an estimated 80 SMS to Twitter every 5 seconds i.e. about 1000 SMS messages per minute [24].

I have used 948 Twitter posts from six Twitter accounts in order to assess the degree to which conflicting reports exist that causes uncertainty. The reason for analysing information from six Twitter accounts only is that I have studied the Mumbai incident about a year after the incident took place. Therefore, it was neither possible for me to collect all Twitter posts relating to the incident due to the limitations of Twitter APIs mentioned earlier, nor was it possible to know which Twitter users were tweeting about the incident. However, different news media referred to some of the Twitter users including *Dina*, and *Vinu*, who twitted regarding the incident [52, 10, 144]. Some of the Twitter users were also ‘mentioned’ (referred to) by other Twitter users. Since, the purpose of this case study is to understand the nature and utility of the information that becomes available on the social networks in the wake of a crisis and emergency, I do not need to collect all the information from all users. I have collected information from among those Twitter users whose accounts remained active (when I attempted to retrieve the data), who appeared to be tweeting from Mumbai and the number of their tweets, made after 29 November 2008, was not more than 3,200 (which, otherwise, would have prevented the Twitter API to retrieve the tweets that were posted when the incident was happening i.e. between 26 and 29 November 2008).

---

<sup>2</sup><http://www.simile-widgets.org/timeline/>

### 3.3.1 Inconsistent and Contradictory Information

It has been found that Twitter users don't always give information based on first-hand experience; in many cases they just relay the messages that they received from different sources (which may or may not be reliable) including the main stream media. With respect to the Mumbai incident, here are some examples of tweets that relay messages from other sources. The second and third tweets refer to CNN News and CNN-IBN and are in contradiction with each other as they differ on the number of deaths of terrorists and they are coming from the same (Twitter) account.

1. At 23:11:53 on 26/11/2008 **mumbaiattack**, a Twitter user, relayed this message from an unspecified source: “#mumbai 78 reported dead >200 injured”.
2. At 03:03:42 on 27/11/2008 **MumbaiAttacks**, a Twitter user, relayed this message from CNN News: “87 are reported dead. 9 of the terrorists are reported dead by a CNN News report. #mumbai”
3. At 03:10:03 on 27/11/2008 **MumbaiAttacks**, a Twitter user, relayed this message from CNN-IBN: “5 terrorists now reported to be shot dead, while 9 are detained. #mumbai CNN-IBN”

Table 3.1 below shows some contradictions found in the tweets.

According to the official statement of the Indian Government, the total number of people injured was 308 [104].

### 3.3.2 Hoax and Rumour

We have seen in the previous section that people spread contradictory information on Twitter. However, the worse thing is that not only do the Twitter users sometimes give contradictory information but at times, they go beyond that and spread rumour or hoax (although possibly without malicious intent). A hoax, apparently created by a Twitter user **mumbaiupdates**<sup>3</sup>, was used in an effort to stop people from reporting live about the military operations against the terrorists [27]. Table 3.2 contains some of the tweets of **mumbaiupdates** that came before and after he created (or at least propagated) the hoax.

---

<sup>3</sup>Note however that **mumbaiupdates** has contributed to the discussion on [27], denying that he was the ultimate source of the rumour.

Table 3.1: Contradictions Found in Tweets about Mumbai Incident

Time	Tweet	Screen_name
26/11/2008 19:58:54	#mumbai - ndtv says 60 dead 200 injured. so far.	dina
26/11/2008 20:08:11	#mumbai 55 dead 190 injured. hostages uk and americans taken	mumbaiattack
26/11/2008 21:15:01	#mumbai 78 dead 200 injured	mumbaiattack
26/11/2008 21:37:48	#mumbai 15 policemen killed so far in intense fighting	mumbaiattack
26/11/2008 23:11:53	#mumbai 78 reported dead > 200 injured	mumbaiattack
26/11/2008 23:12:21	#mumbai 11 members of Police force perish	mumbaiattack
26/11/2008 23:38:04	#mumbai ndtv fm mantralaya. 76 dead 116 injured. 2 terrorists dead. 9 arrested. 2 topcops dead. Chief Minister evasive on who's responsible.	dina
27/11/2008 00:05:56	#mumbai 87 reported dead 200 injured.	mumbaiattack
27/11/2008 01:45:22	87 are reported dead with 185 wounded. #mumbai CNN.com	MumbaiAttacks
27/11/2008 02:08:44	Injured reports rise from 185 to 187 now. #mumbai CNN.com	MumbaiAttacks
27/11/2008 02:59:54	Reports say Mumbai Government has control of situation. 84 dead, 200 wounded, 2 terrorists dead, 9 in custody. #mumbai CNN.com	MumbaiAttacks
27/11/2008 03:03:42	87 are reported dead. 9 of the terrorists are reported dead by a CNN News report. #mumbai CNN News	MumbaiAttacks
27/11/2008 03:10:03	5 terrorists now reported to be shot dead, while 9 are detained. #mumbai CNN-IBN	MumbaiAttacks
27/11/2008 06:31:08	5 terrorists dead, 1 terrorist arrested, and at least 6 still present in the two luxury hotels; Taj and Oberoi. #mumbai CNN-IBN	MumbaiAttacks
27/11/2008 16:01:09	125 reported dead, 327 wounded in the Mumbai attacks. #mumbai CNN.com	MumbaiAttacks
27/11/2008 19:09:02	RT @BreakingNewsOn The death toll from the Mumbai terrorist attacks has risen to at least 131 with 330+ reported injured	mumbaiupdates
27/11/2008 20:19:36	8 terrorists in total have been killed in shootouts so far. #mumbai CNN-IBN on CNN.com Live	MumbaiAttacks
29/11/2008 00:50:48	#mumbai Casualty total: 155 killed 327 injured	mumbaiattack

Table 3.2: **mumbaiupdates**' Tweets Before and After the Creation of the Hoax

Time	Tweet	Screen_name
27/11/2008 03:52:00	Due to military action happening very soon, @mumbaiupdates may have little information to report to protect the rescue operations	mumbaiupdates
27/11/2008 03:54:27	I am not updating on any details about #mumbai operations until further notice to protect the operation	mumbaiupdates
27/11/2008 04:02:59	<b>Indian government is asking that the twitter search page #mumbai be shut down</b>	mumbaiupdates
27/11/2008 04:07:51	or possible clarification: to just stop live updating about the situation pertaining to #mumbai	mumbaiupdates
27/11/2008 04:10:35	ALL LIVE UPDATERS - PLEASE STOP TWEETING about #Mumbai police and military operations.	mumbaiupdates

This hoax was so convincing that even the BBC was fooled to report *Indian government asks for live Twitter updates from Mumbai to cease immediately* with a reference to the tweet made by **mumbaiupdates** on 27/11/2008 at 04:10:35 [52, 27]. However, another Twitter user, **dina**, challenged the authenticity of the message and asked for a proof. When **dina** asked another Twitter user **cool.technocrat** about the source of the information and told him to stop spreading the rumour, **cool.technocrat** responded with the following message with a reference to the BBC website:

@dina read from bbc website <http://tinyurl.com/5al54e> [TweetID: 1026242175, created\_at: 27/11/2008 11:53:27]

**dina** again tried to convince them, saying:

@Kimota please read it carefully - BBC says they got it from a tweet. These rumours have been tweeted all day. BBC is NOT god!!! #mumbai [TweetID: 1026241807, created\_at: 27/11/2008 11:53:04]

Yet some Twitter users were still not convinced, as one of them, **MumbaiAttacks**, tweeted:

CNN-IBN is, unfortunately, yet again reporting specifics as to tactics. They must stop to protect final operations on this assault. #mumbai [TweetID: 1026241807, created\_at: 27/11/2008 11:53:04]

This particular case of rumour highlights an inconsistent behaviour of a trusted and reliable source (BBC) and shows how misleading it can be. However, the rumour about Indian government's request for not broadcasting certain information was a hoax, only because the government did not make such request. Otherwise, the message was not malicious. In fact, the conversation between the terrorists (quoted in Section 2.2.2) proves that some information possibly should not have been broadcast live, as the terrorists were following the media information.

### 3.4 Tool Support: Finding Popular Users by Analysing Retweets and Mentions

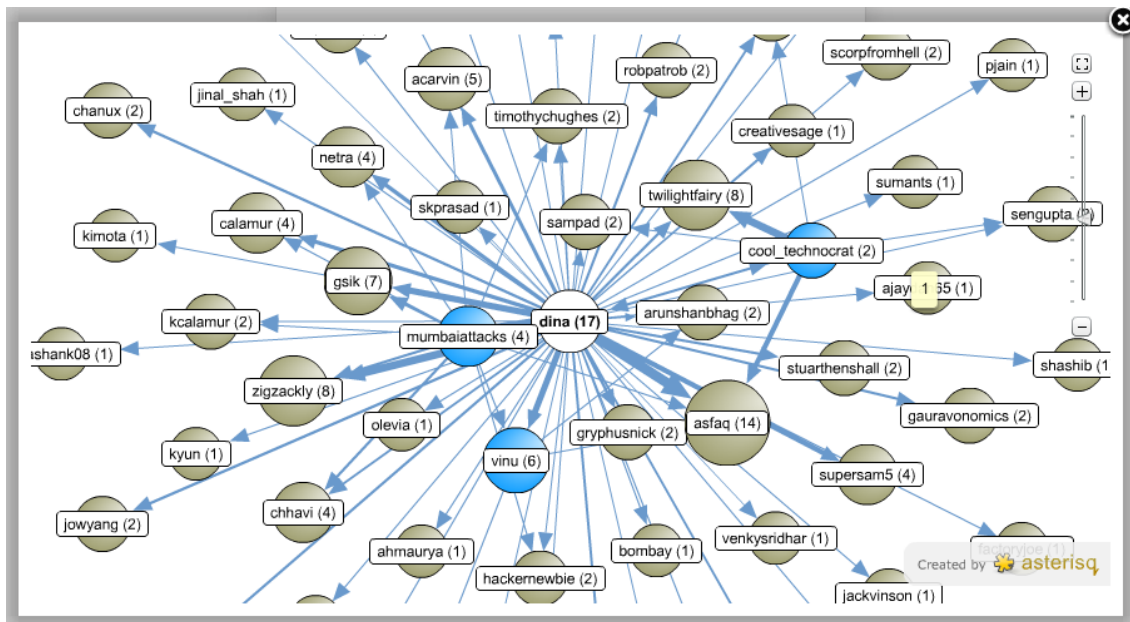


Figure 3.2: Mapping Popularity based on Retweets and Mentions

By observing the incident related tweets (Mumbai Incident), I have found that some Twitter users are relatively ‘popular’ or important in some way, as others frequently mention their names and re-tweet their messages more often. I refer to these Twitter users as ‘popular users’. I have created a support tool (application) to analyse the tweets (that are already saved in my database) in order to find out how many times a Twitter user has been mentioned by other users, so that we can identify those popular users. My application also helps us to visually present

this information using a third party service<sup>4</sup>. We can see from the output of my application, as shown in Figure 3.2, that ‘dina’, the Twitter user who contributed to dispel the rumour discussed in Section 3.3.2, has been mentioned by others the most. From this, it will not be unreasonable to think that most of the times, these popular users may be the main or major sources of information and sometimes, possibly the original source of some information as well. In the context of crisis and emergency, this application is particularly important to all parties including the main stream media for collecting and/or verifying information with the help of these ‘popular’ users after identifying them. Also, in any case, if the Twitter Search tool does not return any/many tweets from those important users, we can collect their tweets from their user-timeline by identifying them with the help of this application. Hence, even though this application tool does not give a direct insight into the uncertainty of the situation, as it only reports on the number of re-tweeting without performing a critical analysis, the output of this application tool is likely to be useful in dispelling/reducing uncertainty in many cases.

### 3.5 Uncertainty in the Aftermath of Haiti Earthquake in 2010

On 12 Jan 2010, Haiti, a Caribbean country, was hit by the most powerful earthquake to strike the country in 200 years [63]. More than 217,000 people were killed by the devastating earthquake, which also left about 300,000 injured and one million people were made homeless [98]. Social networks saw a sudden influx of earthquake related messages as people used these social networks to share and collect information about the disaster and its victims. Facebook received more than 1,500 Haiti related messages per minute [169].

I have conducted another case study on the Haiti disaster by collecting and analysing 306 Twitter status updates from twelve users that were created in the first three days (from 12-15 January 2010) after the quake. I have classified the tweets in different groups based on the main topic of those messages as shown in Table 3.3. According to this classification (i.e. depending on the main topics discussed in the messages), those messages that were reporting earthquake and aftershocks formed the largest group. The classification also reveals that a large number of people were enquiring about others’ welfare and reporting collapse the of different buildings

---

<sup>4</sup><http://asterisq.com>

Table 3.3: Coverage of Topics within Tweet Sample Set (Haiti Earthquake, 2010)

Subject Matter of Tweets	No of Tweets
Quake/Aftershock	44
Finding/Reporting Welfare	39
Building Collapsed	38
General Info (e.g. "...streets are crowded ... people without homes...")	29
Phone / Electricity	24
Rumour	22
Trapped / Injured (Rescue)	21
Local language	14
Dead body	13
Photo	11
Food / Water / Medical Service	9
Out of date information	1
Miscellaneous	41
Total	306

and structures. The reports of the collapse of buildings/structures are particularly important to the rescue services because it indicates where help is most needed. It is also interesting that while 6.9% of the collected messages are reporting people are injured or trapped under rubble, 7.2% of the messages were linked to rumour/hoax (see Figure 3.3). It should be noted that the messages spreading rumours and hoaxes could be classified as such, because the case study was carried out well after the incident when the truth was already revealed.

### 3.5.1 Implicit Disclosure of (Location) Information

The case study also reveals a natural behavioural phenomenon of humans that when people talk, they implicitly give out a significant amount of information about themselves and their surrounding environment, especially their location. Although many people do not specify their location information on their Twitter profiles, I found that people reveal their location information, knowingly or unknowingly, through their tweets. For example, 'RAMhaiti', a Twitter user, did not provide his location information on his Twitter profile. However, some of his tweets shown in Table 3.4 clearly indicate that he was in Haiti at the time of earthquake.

It also appears from the tweets that at least ten out of these twelve users twitted from inside Haiti and one of these users reported the earthquake within three

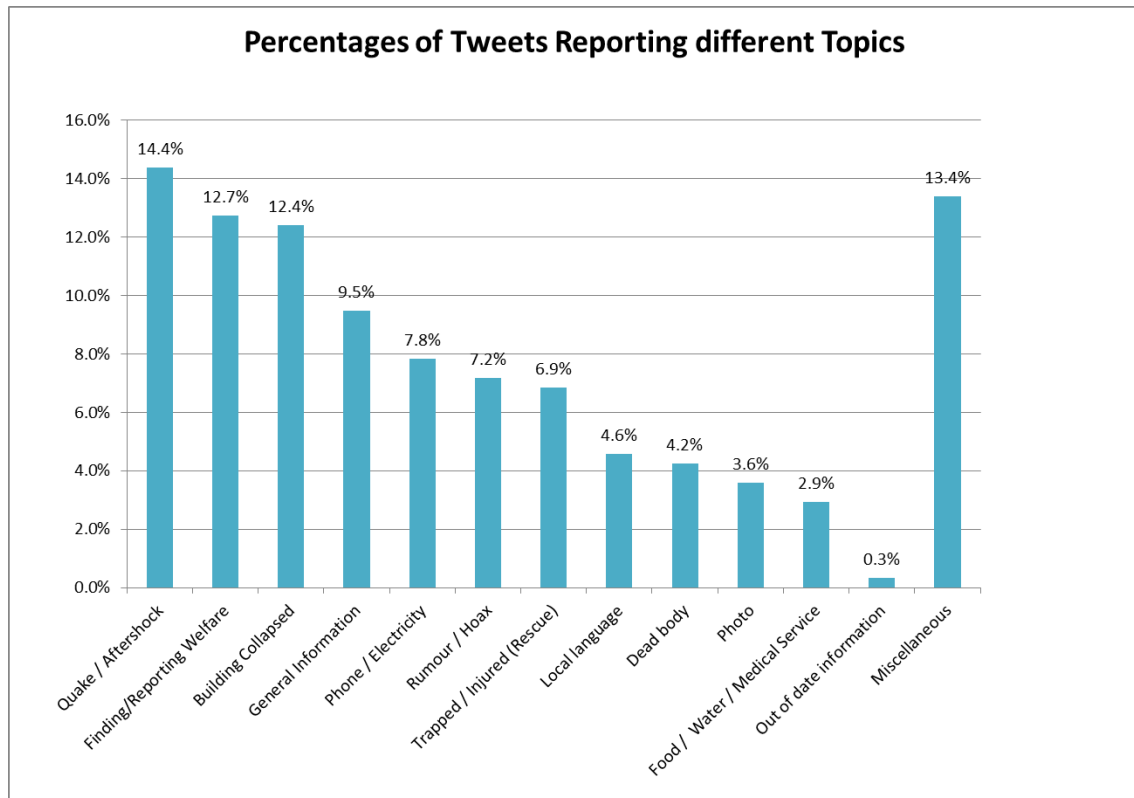


Figure 3.3: Classification of Tweets based on the Main Topics of Messages

Table 3.4: People Reveal their Location Implicitly on Twitter

Time	Tweet	Screen_name
13/01/2010 00:45:35	another aftershock..people are screaming and freaking out down towards the stadium..much singing and praying in large numbers	RAMhaiti
13/01/2010 01:45:58	It's 8:44PM and we're still getting aftershocks!!I can hear people gathered in the distance singing prayers	RAMhaiti
13/01/2010 01:58:06	My mom just showed up at the Oloffson..sigh of relief!!	RAMhaiti
13/01/2010 17:28:33	we're back on line..went to St Gerard Church..people are trapped in the school..others are dead in the rubble	RAMhaiti

Note:

1. 'Oloffson' refers to Hotel Oloffson, Port-au-Prince, Haiti
2. 'St Gerard Church' refers to St. Gerard church and school in Port-au-Prince, Haiti



minutes of the incident. The earthquake hit Haiti at 21:53:10 on 12 January 2010 and **FredoDupoux**, a Twitter user, tweeted “*oh shiet heavy earth quake right now ! in haiti*” at 21:56:20 on 12 January 2010.

Table 3.5: Tweets Containing Valuable Information

Date and Time	Tweet	User ID
13/01/2010 13:57:18	C’EST URGENT! To any1 in MontJoli turgeau area plz go to the Neptune house, Jean-Olivier is caught under the rubbles!!!!	Freichrisha
13/01/2010 19:38:32	RT @tikrezi: Chantal Landrin is stuck under the rubble at a house in Turjo! someone please help her. #Haiti	Yatalley
14/01/2010 11:26:51	Jacmel has had much destruction,school kids caught in collapsing buildings..Jacmel situation is bad	RAMhaiti
15/01/2010 04:08:59	****URGENT**** LYCEEE FRANCAIS TO RESCUE GISCARD LEFEVRE ...HE’S ONLY 8....HE’S ALIVE AND STUCK UNDER THE RUBBLES.....	Freichrisha

### 3.5.2 Windows of Uncertainty

Some of the tweets, as shown in Table 3.5, appeared to contain valuable information relating to people being stuck under rubble, or children caught in collapsing buildings. On the contrary, some other tweets appeared to spread rumours and hoaxes that would make it difficult to believe the messages or make decisions based on them, such as the presence and degree of local flooding. Table 3.6 includes some tweets showing that some people were spreading a hoax about free flights to Haiti for doctors and nurses, while others were trying to stop that hoax. Figure 3.4 shows the window of uncertainty (period of uncertainty) that was caused by some tweets that were spreading the rumour of flooding after the earthquake. The window of uncertainty lasted for about 19 minutes.

We have also seen in Section 3.3.2 that hoax created a window of uncertainty during the Mumbai incident, which lasted for at least 7 hours 50 minutes (from 27/11/2008 04:02:59 to 27/11/2008 11:53:04).

It should be noted that I only studied those tweets written in English, resulting in about 5% of the collected tweets being excluded (as they were written in local

Table 3.6: Tweets related to a Hoax that spread after the Earthquake in Haiti

Date and Time	Tweets	Screen_name
14/01/2010 02:02:09	RT @mrprodg: AMERICAN AIRLINES is taking doctors and nurses to Haiti for free. Please call 212-697-9767. Spread the word.	KlausAK47
14/01/2010 02:44:48	RT @grosdim: American Airlines is taking doctors and nurses to Haiti for free. Please call 212-697-9767. Spread the word	FredoDupoux
14/01/2010 19:55:01	RT @RAMhaiti "American Airline and Jet Blue Are NOT Flying doctors and nurses into Haiti for free. Do NOT call them."	eveblossom
15/01/2010 05:27:10	RT @InternetHaiti: Twitter hoax spreads rumors of airlines' free flights to Haiti <a href="http://bit.ly/4nKv01">http://bit.ly/4nKv01</a> #Haiti	FutureHaiti
15/01/2010 16:26:30	Jet Blue is giving free flights to doctors and nurses to Haiti CONTACT HT CONSULATE IN NY : 212-697-9760 Contact@haitianconsulate-nyc.org	freichrisha

languages such as French and Haitian Creole). If processed, those messages might have given more important information.

### 3.6 Summary

The case studies highlight the fact that while there is a large volume of potentially useful information available on the social networks, a large amount of noise, misinformation and possibly disinformation comes with that too. Two very recent hacking incidents also give the same message about information sources on the internet as false information was posted on the Twitter accounts of AP and BBC by hackers [99, 31]. This makes the use of information from social networks difficult, especially in the case of crisis and emergency. Emergency controllers will have to deal with multiple, partial and possibly conflicting reports. However, the good thing is that in both case studies, the content of the tweets (about some issues) did converge over a period of time and thereby removed some uncertainties.

Therefore, the aim of this research is to investigate how the varying provenance of the data can be tracked and exploited to prioritise the information presented to a busy incident controller and to synthesise a model or models of the situation that the evidence pertains to. These models of the situation will give the user the ability

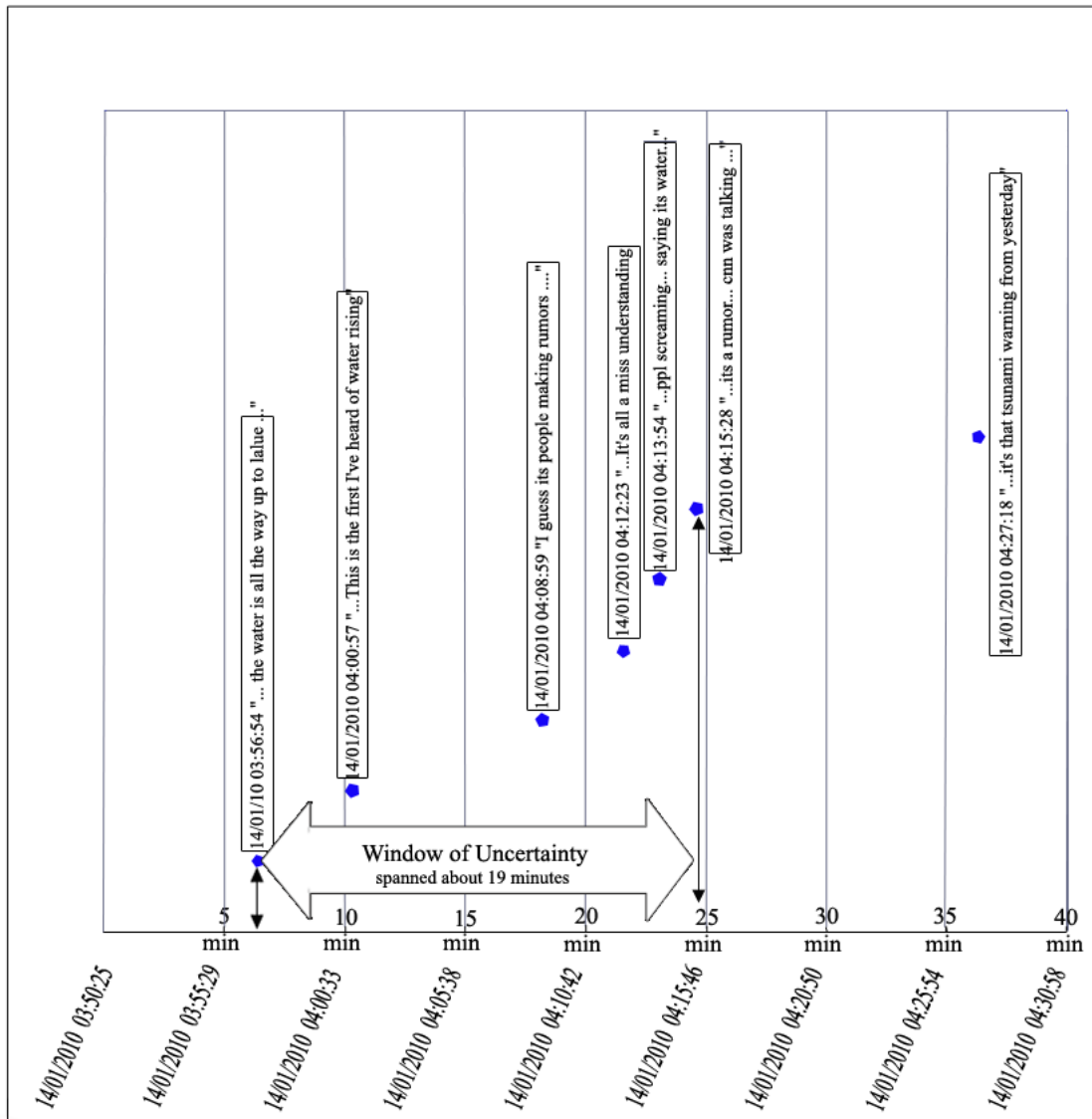


Figure 3.4: Rumour created Uncertainty about Flood for about 19 minutes

to utilise open-source information with an increased awareness of likely provenance.

## CHAPTER 4

# Provenance and Quality Factors Affecting Trust in Information

Information from open-sources, including social networks, has a potential value for crisis and emergency management as was shown in Chapter 3. However, the information from such sources can only be used after filtering out noise, misinformation and disinformation. Therefore, it is essential to have a thorough knowledge and understanding of what influences the trustworthiness of information.

People's trust in information depends on various factors as outlined in many research articles [103, 17, 15, 50, 49, 56, 140, 57, 70, 77, 96]. An exhaustive literature review, presented in [49], provides a list of factors that influence how end-users make decisions regarding trusting information. These factors include:

- Topic (trust in a resource is topic-dependent)
- Context and criticality (context determines the criteria by which a user judges trustworthiness)
- Authority (source identity and competence influence trust)
- Direct experience (reputation leads to trust)
- Recommendation (referrals from other users provide indirect reputation)
- Bias (biased sources may convey misleading information)
- Motivation or incentive (information may be more believable if there is an incentive for a resource to provide accurate information)
- Agreement (corroboration influences trustworthiness)

- Age (time of creation/lifespan of time-dependent data indicates its validity)
- Appearance (user perception of a resource affects the trust of the content)
- User expertise (expert users may make better trust judgements on a resource's content)
- Limited resources (absence of alternate resources may result in trusting imprecise information)
- Likelihood (probability of content being correct, in light of everything known to the user)
- Recency (content, associations and trust change with time) and
- Provenance

However, all the factors/properties that influence people's trust in information can be broadly categorised into three main groups: factors relating to the provenance of information, factors relating to the quality of information itself and factors relating to the information consumer [103]. A detailed discussion on the main factors from all three categories, which are predominant in both research and application domains, is presented in the following sections.

## 4.1 Provenance Factors Affecting Trust in Information

Provenance, also referred to as lineage and pedigree [127], of information refers to the source of information such as who gave (or produced) the information, the derivation history of information, what data was used to generate it, and also finding the trail of how the information has passed from one source to the other and how it has been changed. A recent study shows that one of the main factors that influence the trust of users in web content is provenance [49]. Provenance of information has direct and indirect links with security properties, especially with authenticity and integrity of information and access control [80]. Thus, provenance information helps to assess the quality of information (correctness, authenticity, integrity, etc.) and thereby, helps to determine the level of trust that can be attributed to it [56, 140]. I propose to use measures of provenance in order to score open-source data and provide a method for filtering through the various world views resulting from uncertainty; more priority can be given to the information from more trusted sources. (However,

security of provenance information is a critical issue, since there is a potential risk of having the provenance information tampered by malicious agents that will ruin the purpose of keeping provenance information.) There are many factors proposed for judging provenance, sometimes explicitly and sometimes implicitly, designed to provide evidence or measures of trustworthiness. The selection of factors may depend upon context, since their importance and reliability as a provenance indicator may vary depending on the situation and their vulnerability to compromise and the likelihood of a compromise (malicious or accidental) taking place. The results of a survey into factors commonly in use and the potential points of their vulnerability are outlined below.

#### 4.1.1 Identity of Informer

Identity provides a base for trustworthiness, risk assessment, and provenance [161, 30]. If we know the identity of the informer and other demographic information related to them, then it may help us to understand their motive. Generally, when we trust a person we believe the information s/he provides. So establishing identity can be essential to underpinning trust. Various pieces of information contribute to an identity, and could include name or pseudonym (user name) which remains consistent over time and can be linked to an individual, phone number, email/IP address, age, education, profession and membership of social groups. It has been argued that adult people are more trustworthy than children and adolescents [139]. We may believe an old lady more than a 13 year old boy unless there is a doubt that someone might have masqueraded as an old lady by changing his/her voice or the (real) old person has a dementia. There may be more reasons to act upon a tip-off received from an off-duty police officer than from a member of the public. Similarly, information from someone with a track record in providing quality information may be more believable than information provided by a previously unknown entity. It is clear that identity itself does not signify anything in relation to the person's trustworthiness unless we know more attributes (which we might refer to as competence in part). For example, we have received messages from two email addresses e.g. [abc@defence-administration.uk](mailto:abc@defence-administration.uk) and [xyz@defence-administration.uk](mailto:xyz@defence-administration.uk). Since we do not have any more information about the sources (except their identities), so we do not know whether they are the two most senior intelligence officers or two blue-collar workers. Hence, we cannot treat their information as very dependable despite the fact that we have the identities of the informers (their email addresses are

their identity). Having said that, we may treat the messages (that are not phishing emails and are really) coming from the server of “National Defence Administration” as more reliable than the messages coming from a Hotmail account even though the messages were sent by blue-collar workers of “National Defence Administration”. This is because of two reasons.

Firstly, someone sending messages from “defence-administration.uk” is more traceable and therefore, they are less likely to be malicious.

Secondly, the server that belongs to the “National Defence Administration” is expected to be more secure against attacks and Identity Theft. Forged identity and credentials are widely used by scams to steal money from people’s bank accounts through Phishing [110]. The use of forged identity is also troublesome for social networks. For example, the Twitter accounts of two reputable mainstream media organisations (AP and BBC) have been hacked very recently [99, 31]. A false message saying “Breaking: Two Explosions in the White House and Barack Obama is injured” was posted on the Twitter account of AP (Associated Press) after it was allegedly compromised through Phishing [99]. The BBC Weather account on Twitter was also hacked and tweets relating to the current conflict in Syria were posted from the compromised account [31]. A similar kind of attack took place on the US president Barack Obama’s Twitter account as some of the tweets on his account were not made by him [13].

Since establishing the true identity may be a challenging task, information received from agents, especially with virtual identities (email address, Twitter Id, etc.), should be used with caution as the identities are vulnerable to forgery.

#### 4.1.2 Location of Informer

The location from which an informer reports an incident may also have a bearing on the believability of information; we might always give more importance to the information received from an eyewitness as we believe it to be more accurate. We can use location as a method for determining whether claims to eyewitness accounts are credible; we can rule out informers who are not within a given radius of the incident. However, collecting location and other private information pertaining to information providers may be extremely difficult because of legal and privacy issues (unless they give consent to providing it), and there is a potential risk of having location information faked. A researcher at the University of Illinois at Chicago has demonstrated such attack using only 9 Perl statements [73]. There is also software

for faking the location of mobile phones which is easily accessible in the market at a very low price e.g. [Fake-A-Location](#)<sup>1</sup>. Although IP addresses could be used to find an approximate but true location of an informer, IP address can also be faked with readily available software e.g. [Hide My IP 5.2](#)<sup>2</sup>, [Hide My IP Address](#)<sup>3</sup>.

It should also be noted that a person can witness an incident from the top of a building which is reasonably far from the incident location, while another person cannot be an eyewitness for being behind a wall or other obstructions despite being much closer to the incident location. Hence, deciding on a radius (distance of an informant/eyewitness from incident location) that can be used to assess the credibility of someone's claim to be an eyewitness is a challenging task and is a topic for future research.

### 4.1.3 Reputation

We can predict someone's behaviour with a known history (reputation) to a certain extent, while it is almost impossible to predict their behaviour with a little or no history about them. So, we may trust an informer with a good reputation more than another with a bad reputation (or no reputation at all). However, there is a risk associated with using reputation for evaluating trustworthiness because there is no guarantee that an informer with a very good reputation will not give wrong information [49]. A trusted source may suddenly (willingly or inadvertently) issue a false statement, or a typically distrusted source may post information that is trustworthy [7]. Therefore, this factor will not be reliable in situations when a source has purposefully built up a positive reputation specifically to act as cover for the time at which they wish to act maliciously (equivalent to an insider threat whose reputation had been built up over time and who suddenly steals some data). Misinformation from previous apparently-reliable source may be subtle and hard to detect immediately (as in the case of Advanced Persistent Threat [142]). Hence, we need to be cautious while using reputation as a measure of trust.

### 4.1.4 Popularity

When a message appears on a social network (e.g. Twitter, Facebook, etc.) informing about an incident, many other people repeat that message and many people

---

<sup>1</sup>[www.excelltechmobile.com/](http://www.excelltechmobile.com/) (Accessed on 19/12/2010)

<sup>2</sup>[www.hide-my-ip.com/](http://www.hide-my-ip.com/) (Accessed on 18/12/2010)

<sup>3</sup>[www.hide-my-ip-address.com/hideip/](http://www.hide-my-ip-address.com/hideip/) (Accessed on 18/12/2010)



start following that first informer for more updates. Thus, a number of popular users are found on social networks that frequently provide a large amount of timely information about incidents. Of course, it is questionable the degree to which popularity should be considered a measure of trustworthiness, as it is entirely possible that many are following someone who is incorrect in their assertions. However, some believe that popularity must mean that either the source is providing accurate information or is infamous for something related to the incident, as there appears no other rationale explanation for their popularity (they are not celebrities). Who is being mentioned how many times can be worked out from Twitter messages and a corresponding score can be assigned (positive or negative) to each popular user according to their popularity. It is also possible to capture the average number of times someone's message gets repeated by others. The two, when combined, could give an estimate of popularity, should such a factor be considered important to provenance in a given situation. The heuristics on how different scores may be combined to calculate the overall score is described in Section 5.1.5.

#### 4.1.5 Context/situation, Interest and Ethics

People's trustworthiness varies depending on the situation or context [17, 123, 15]. It is easy for someone to be unbiased and trustworthy in a situation when his/her own interest is not concerned. We may doubt a trader's word when s/he says that s/he is not making any profit by selling a particular product. However, we will have no reason to doubt the same person when s/he reports an accident. However, intention itself is sometimes context dependent and driven by ethics. For example, a journalist who is also a share holder of a company may write a true report about the company because of his/her ethics, despite the fact that the report will cause the share price to fall.

However, in reality, we cannot judge someone's trustworthiness based on their intention or ethics as there is no practical way of reading someone's mind and knowing their intention. It is necessary to use their history/reputation to know about their ethics. So this factor will be vulnerable in the same way as reputation (discussed above).

#### 4.1.6 Social Relation

A person,  $X$ , is likely to trust another person,  $Y$ , to a certain degree if  $Y$  is trusted by many other people who are trusted by  $X$  even though  $X$  does not know  $Y$  [70, 77]. This transitive nature of trust is also seen in social networking websites (virtual society) e.g. Facebook, LinkedIn, etc.. However, the degree of transitivity of trust depends on other factors. For example, a cousin  $A$  of  $X$ 's mother, and a friend  $C$  of  $X$ 's friend  $B$  are unlikely to be trusted by  $X$  to the same degree. This is because the trust is rooted in two different ways (cousin of mother and friend of a friend respectively). Social relations can be easily manipulated with a little use of social engineering and masquerading. For example, Alice and Bob are very good friends and their houses are on the same street. On an evening a scam, Mallory, knocks on Alice's door and identifies himself as Bob's brother. Mallory asks Alice whether he could sit in her sitting room while he is waiting for Bob and it is very cold outside. There may be a good chance that Mallory will get access to Alice's house and steal her mobile phone. This manipulation is even easier in social networks as there is a significant amount of private information publicly available on the internet which may facilitate social engineering. It is noteworthy that if Charlie identifies himself as Mallory's brother and wants access to Alice's house with a similar excuse while Alice does not like/trust Mallory for some social (or personal or other) reasons, then it is likely that Charlie will not get access to Alice's house due to his relation with Mallory. However, trust based on social relationship is very subjective; because another person (despite being in the same situation as Alice i.e. having a bad impression of Mallory) may accept Charlie's request based on the opinion that 'not all are alike' i.e. Charlie may be a good/pleasant person unlike his brother Mallory. Hence, it may not be a reliable factor for assessing trust, in all cases.

#### 4.1.7 Corroboration

When the same information comes from many different and unrelated sources we tend to believe the information, even though the sources aren't very trusted or they have no previous reputation [7, 30]. But, if it is found that the sources are related to each other then it may cast doubt on the information. ENISA's survey shows that most web users trust the content of a website because it is found on many other websites [96]. However, the potential risk in establishing trust based on corroboration is that it may not be possible to unearth the fact that some websites

are related to each other and are maintained by the same crime syndicate. For example, it was found that false rumours have been spread using web 2.0 applications in order to manipulate stock prices [97].

#### 4.1.8 Competence

Trustworthiness of information also partly depends on the competence of the information provider. If someone does not have adequate level of knowledge and expertise (competence) to securely generate/collect some information in a specific context then, the information provided by him/her may not be trustworthy and reliable [50]. Hence, if someone reports an incident and says that some of the casualties are suffering a brain haemorrhage then we will think of either of two possibilities:

1. Some casualties are possibly suffering a brain haemorrhage and the informer is possibly a doctor/nurse/paramedic.
2. While there may be some casualties but the information of the brain haemorrhage is possibly baseless as (we think) the informer does not have the necessary knowledge, expertise or skills to diagnose a brain haemorrhage.

Likewise, if a fireman recognises and reports a shooting incident as an organised terrorist attack then we cannot be certain about it. However, this factor may be vulnerable if a competent person is biased, since his/her judgement may be biased or prejudiced.

#### 4.1.9 Conviction/Certainty

When someone receives a piece of information, they can believe or disbelieve it to a certain degree. Likewise, a person may not completely believe a message even though s/he passes it on. People, despite being a sender of a piece of information, believe or disbelieve the information to a varying degree. Therefore, it is important to ask information providers as to how certain they are about the veracity of the information they are providing, especially when they have received that information from others. If the source itself is doubtful about the veracity of a piece of information, then we should automatically give less priority to that information provided that we believe the source is honest. Since, source certainty i.e. the conviction of a source about the veracity of information solely depends on the source's honesty and personal view, it will be very risky for us to make a decision based on this

factor alone; as this is an extremely vulnerable factor [46]. A source that expresses a higher certainty about the veracity of a piece of information which turns out to be untrue may be penalised with a diminished reputation or trustworthiness.

## 4.2 Information Quality Factors that Influence Trust

Information quality can be defined as an assessment or measure of how fit an information object is for use [103]. This notion of “fitness for use” is central to several information quality discourses and is apparent in numerous research articles [153, 72, 122, 18, 69]. A crucial question which surfaces in most information-quality research literature is, what are the dimensions/factors that define information quality and thus lead to believing in that information [103, 153, 72, 122, 18]. The following sections outline those factors (Information Quality Factors) that make information trustworthy.

### 4.2.1 Security of Information

Quality of information deeply depends on information security (security of information and its communication infrastructure). The quality of information has to be ‘good enough’, if anyone wants to use that information for a specific purpose. A piece of information cannot be reliable and thus, cannot be used if that information is deemed to be unauthentic and/or tampered, even though the information appears to be crucial for a specific task. It is noteworthy that security makes both information and provenance of information credible and reliable. However, security and provenance of information are interdependent and sometimes they are called ‘symbiotic’ [83].

### 4.2.2 Freshness of Information

Freshness or timeliness is one of the main factors that determines the quality or believability of information [15, 57]. It is essential to know when the information was published or message was sent in order to judge how fresh the information is, and assign a probability of correctness or trustworthiness accordingly. When we receive conflicting information, freshness may play a part in deciding which information to base a decision on, as it is likely that the freshest (the latest generated) piece of information provides some previously unknown/undiscovered information (although

the decision is likely to be based on more than just a freshness factor). In some cases, there may be a significant time-gap between sending information and receiving it at the other end. As a result, old information will appear as fresh and correct information. If we can ensure that each message published in the mash-up comes with a time-stamp then it will be possible for us to know whether the information is old or fresh. For example, mobile phone network operators inform the recipients of the time when a text message was sent to them by adding a time-stamp to each message when they receive it from the sender [133].

### 4.2.3 Correctness and Accuracy

Accuracy of information has been described as one of the factors that makes information trustworthy [153, 122]. Another factor that influences trust in information is the correctness of information (free from error) [108]. For example, someone publishes the following report on a blog or, on Facebook:

*“Literacy rate is very high in Mars. 85% people (excluding those who are underage) living in Mars cannot read and write, while 20% of the total population is highly educated. Another interesting fact about Mars is that most of its educated people write with their left hand. Only 70% of them write with the right hand.”*

Because the report has noticeable errors, it will be difficult for people to trust this information. Likewise, inaccurate information (e.g. inaccurate driving directions) leads to confusion, which results in diminished reliability of that information.

### 4.2.4 Completeness

Incomplete information causes confusion which leads to distrust. For example, it has been found that when holiday makers look at a travel agent or tour guide’s website and find that the website does not provide complete information about a particular topic, then they find it unreliable and do not want to buy any package from that agent [8]. Incomplete information is also an indication of bias [107] and information from a biased source is not well trusted [49]. When complete information is not provided then even if people believe in the available information, yet they start questioning the honesty and reliability of the source. For example, HM Revenue and Customs (HMRC) recently published pictures of the 32 tax evaders to name and shame them, claiming that those people were the “top tax cheats” [100, 25].

Since the published list was not complete, people who read the report made various comments (see Figure 4.1) showing lack of trust [100].

It should be noted that although Figure 4.1 shows the views of only eight (8) readers, however, their views were supported (thumbs up) by other 736 people while only 27 people did not approve (thumbs down) those views.

#### 4.2.5 Objectivity

One of the most important factors that influences trust in information is its objectivity. A report includes less objective information means it contains more subjective information. If anything is subjective, then (by the definition of the term ‘subjective’) that thing (e.g. information) is “influenced by or based on personal beliefs or feelings, rather than based on facts” [34]. Hence, the reliability of subjective information is always questionable. It has been argued in [135], giving a reference to [54], that objectivity gives an indication as to how much a news or information item has strayed from trust and fairness.

### 4.3 Factors related to Information Consumer that Influence Trust

Trust in information not only depends on the quality of information itself and the trustworthiness of its source, but also on the information consumer. People’s disposition to trust and their propensity to taking a risk play an important role in deciding whether to trust a piece of information or not. Therefore, any decision support system that evaluates the trustworthiness of information should allow its users to incorporate their own opinions or choices, if they want to, in the process of deciding whether to trust a piece of information or not.

#### 4.3.1 Disposition to Trust

Trusting someone or something also depends on people’s personality [114]. Some people are more likely to trust while there are others who do not want to trust easily. However, it has been argued that if trusting means believing others when there are no clear-cut reasons to disbelieve and not being foolish, then high-trusting people (people who trust more) are not necessarily more gullible than low-trusting people [115].

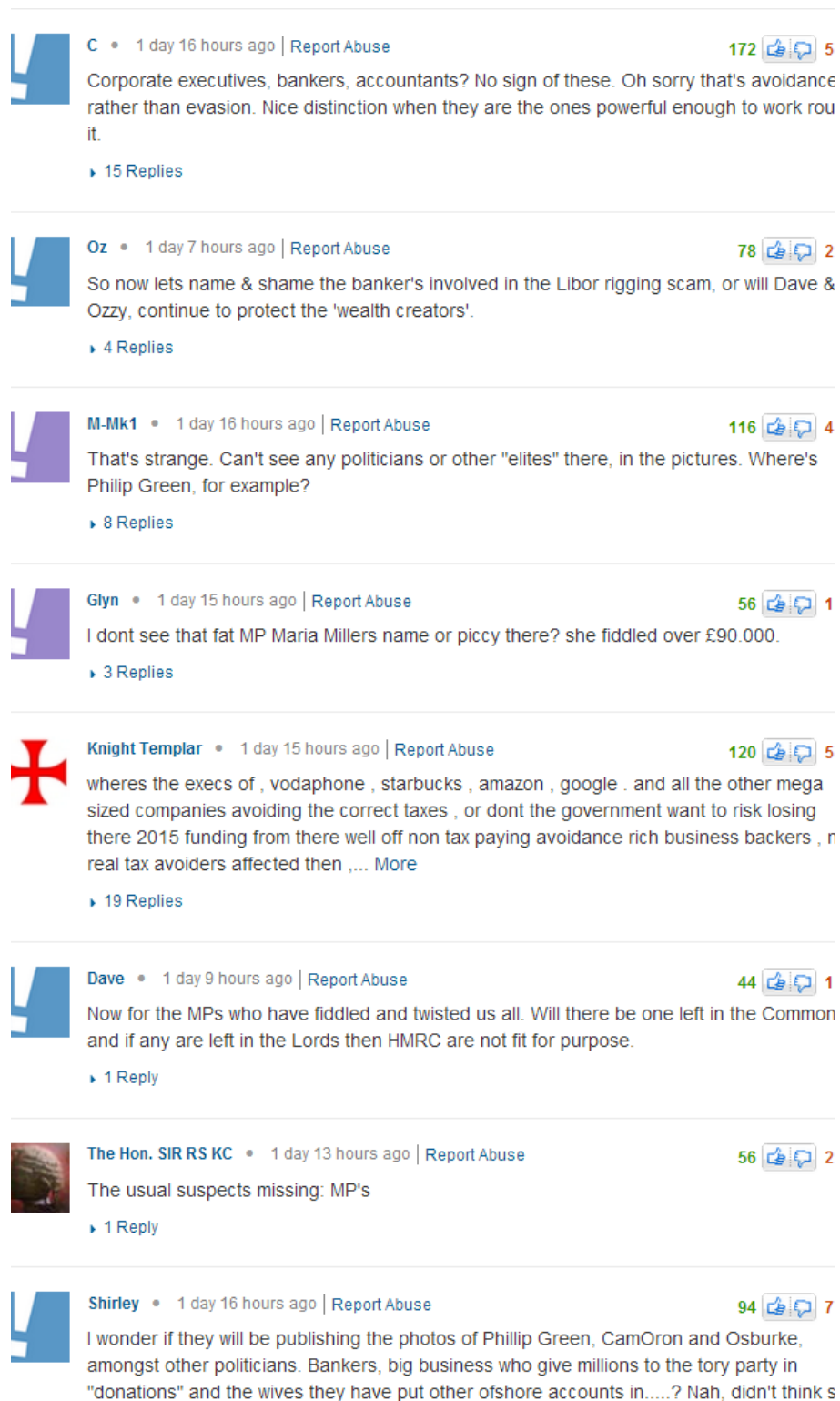


Figure 4.1: People's Response to an Incomplete Information on 'Top Tax Cheats' [100]

### 4.3.2 Propensity to Risk Taking

If someone has a doubt about the veracity of some information after considering all the (other) factors that influence trust (in information), then that person performs a risk analysis before acting upon that information. If the anticipated risk associated with acting upon a piece of information exceeds the level of risk some is happy to take, then s/he will not act upon that information. However, people's propensity for taking a risk is domain-specific or context dependent, i.e. they are not always risk-averse or consistently risk-seeking in all situations [158]. Risk propensity is also strongly linked with personality, age and gender [101]. The following example illustrates that the perception of risk and risk-propensity are context dependent. If the police receive a report of a fight between a few (not many) school children, they may not respond to that report considering the event as low-risk. On the other hand, if the police receive a report that a primary school child is chasing a few of his classmates with a real pistol in hand (not clear whether the pistol is empty or loaded), they are likely to respond to the report immediately as the associated risk is very high.

## 4.4 Vulnerabilities of the Provenance Quality and Trust Factors

While the information quality and provenance factors, discussed in Section 4.1 and Section 4.2, certainly indicate the trustworthiness of information, all of them have certain vulnerabilities. The vulnerabilities of these factors have been presented in Table 4.1 so that they are considered when the factors are used to evaluate trustworthiness of information. With regard to the factors associated with the information consumers that affect trusting a piece of information, it is worth noting that the consumers' disposition to trust and their propensity to taking a risk may have both positive and negative impact on decision making.

## 4.5 Summary

This chapter discussed the factors that influence trust in information which may be helpful in developing an automated system for evaluating trustworthiness of infor-



Table 4.1: Vulnerability or Limitation of Factors that Influence Trust

<b>Factors</b>	<b>Limitation or Vulnerability</b>
Identity of Informer	Identity Theft (use of forged identity and credentials)
Location of Informer	Collecting location and other private information may be extremely difficult because of legal and privacy issues. There is a potential risk of getting faked location information.
Reputation	Agents with a good reputation can betray/defect e.g. insider threat. Reputation cannot be maintained for anonymous agents.
Popularity	Syndication of criminals may make it difficult to use this factor successfully.
Context/situation, Interest and Ethics	People's trustworthiness varies depending on the situation or context. However, there is no way of reading someone's mind and knowing someone's intention.
Social Relation	Social relations can be easily manipulated. This manipulation is even easier in social networks as there is a large amount of private information publicly available on the net which will facilitate social engineering.
Corroboration	It may not be possible to unearth the fact that some of the corroborating sources (e.g. websites) are linked to each other and are maintained by the same crime syndicate. For example, it was found that false rumours have been spread using web 2.0 applications in order to manipulate stock prices [97].
Competence	Knowing competence is difficult as it depends on knowing the true identity of the source. Moreover, a competent source may be biased or prejudiced.
Conviction/ Certainty	Knowing this information in a non-interactive system will be difficult.
Security of Information	There is no such system that is one hundred percent secure and there is always a possibility (low or high) for a system to be compromised.
Freshness of Information	There may be a significant time-gap between sending information and receiving it at the other end. While a time-stamp could distinguish between old or fresh information the time-stamp itself can be forged.
Correctness and Accuracy	It is difficult to assess the correctness and/or accuracy of information during crisis and emergency.

mation and making decisions based on that information during crisis and emergency situation. All the factors presented in this chapter are important, found in numerous research articles and also used in different application systems. While all these

provenance and information quality factors are useful in the evaluation of trustworthiness of information, thus relevant to the work presented in the following chapters, almost all of the them have certain vulnerabilities (as shown in Table 4.1) that we need to be aware of while using these factors. However, all of these trust factors may not be equally important or applicable in different situations. Their importance will depend on how relevant they are in a given context as illustrated in Section 5.1.3. Moreover, it has become clear from the discussion in previous sections that extracting and measuring different provenance and quality factors of information is not an easy task. Some of these factors (e.g. Context/situation, Interest and Ethics, Competence, etc.) are very difficult to extract or measure, while some other factors such as Location of Informer, Reputation, Corroboration, Freshness of Information and Popularity may be worked out relatively easily. Therefore, Location of Informer, Reputation, Corroboration, Freshness of Information and Popularity may be used in the decision support system (presented in Chapter 5).

Some of the trust factors are dependent on other factors to certain degrees. For example, correctness and accuracy of information is likely to be affected by the competence of the information provider. In other words, correctness and accuracy of information may be an indication of the competence of the informant. It has also been discussed earlier that identity forms the basis for other factors such as Reputation, Social Relation, Competence, etc. If someone's identity does not remain persistent over time, other trust factors that depend on identity may not be useful.

There are some other factors that influence trust in information. These factors are related to the information consumer, instead of the source of information. Although, the information consumer related factors cannot be used to evaluate the sources of information, they highlight the need for a trust based decision support system to allow its users to customise its output according to their own view and risk appetite. However, allowing the user to incorporate his/her own view can have both positive and negative effects on the decision process. Since the primary reason for using a decision support system in the face of uncertainty during crisis and emergency is the lack of reliable information and situation awareness, it may be wise to restrict the users from overturning the system's suggestions altogether.

## CHAPTER 5

# Decision Support System: High-Level Design and Algorithm

An intelligent system needs to be designed that can evaluate the trustworthiness of information based on some given criteria/preferences and help people, particularly the emergency responders, to make an informed decision. Provenance of information and other factors that influence trust in information (as discussed in Chapter 4) can be used in the Decision Support System (DSS) to evaluate the trustworthiness of information. This chapter presents a novel design and function of such a decision support system. This novel system (DSS) presents one or more possible scenarios of a situation by evaluating the credibility of information, performing knowledge-based consistency analysis of information and clustering consistent information together. Each possible scenario of a situation is referred to as a *world view*. That is, a *world view* is collection of consistent messages, which provides a possible picture of a situation constructed from the evidence found in the messages (harvested from open-sources). Ideally, a world view is not restricted to provide a picture of any specific number of incidents. It (a world view) is expected to provide as much accurate picture of a crisis and/or emergency situation as possible, while a crisis or emergency situation may be caused by a single or a series of incidents (e.g. armed miscreants opened fire on a moving vehicle. The vehicle lost control and crashed into a nearby shop before its engine caught fire.)

If the system encounters multiple possible world views, then it presents the world views in the order of their (policy based) credibility score.

## 5.1 Top-Level Design and Architecture

Figure 5.1 shows the design of the decision support system that makes a novel use of open-source information and produces one or more possible world views supported by evidence found in the harvested information. The first task of the decision support system (as shown in Figure 5.1) is to harvest information or messages from selected open-sources using its policy based filtering component. However, information collected from multiple sources are very likely to be structured in different formats, even if they are not totally unstructured. Therefore, structuring such information into a single and suitable format is a prerequisite for making any good use of such heterogeneous information. Hence, the next task of the DSS is to encode the harvested data into a single structure which is suitable for disaster related information and its analysis. Each piece of (structured) information is then given a score (according to a policy based on the information provenance and quality factors) before stored in a data repository. One of the major components of the DSS is the “Consistency Analysis & Conflict Resolution” unit, which separates contradictory information and clusters all consistent information together. This is how different ‘world views’ are created. However, these world views do not have an overall score associated with them. The Combiner Function works out the overall score for each world view based on individual message scores before the system presents the ordered views of the possible world. Various components of the system shown in the architecture are discussed in more detail below.

A summary on the operation principles and implementations of different components of the DSS, including which of the components require manual processing and which of them are automatic, etc. is given in Table 8.1. Table 8.1 also outlines the input and output (what the input/output data is, their data types and/or data source) of each component of the DSS.

### 5.1.1 Information Source Filter

Whilst the system can collect open-source information from any source and any number of sources (limited only by the data handling capability), it is expected that user organisations of such a decision support system may wish to filter out some sources for various reasons including relevance, source reputation, noise reduction, corporate policy, limited data handling capacity, etc. The Information Source Filter (tagged with number 1 in a circle in Figure 5.1) will simply provide a mechanism

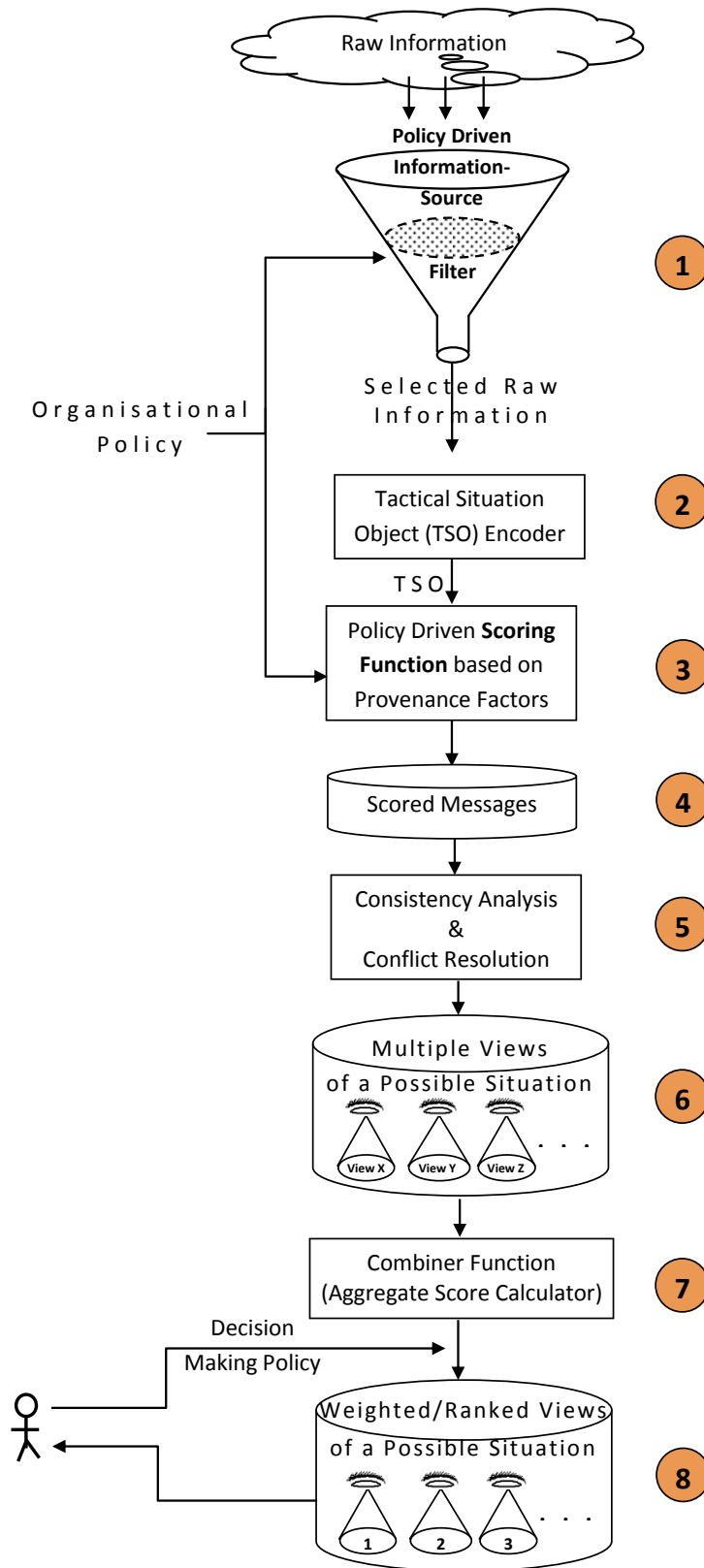


Figure 5.1: Data Flow in the System Architecture

for either subscribing to a set of known information sources, or conversely blocking inputs from a set of known information sources. This forms one component of the *Organisational Policy*.

### 5.1.2 Tactical Situation Object (TSO) Encoder

Unstructured data is never suitable for analysis, both manual and automatic. Understanding the semantics of a message (written in plain-text) automatically is impossible without using Natural Language Processing (NLP) techniques. However NLP technology is not matured enough to serve this purpose [75, 136]. It is, therefore, essential to convert unstructured data into a suitable structured format. Data of heterogeneous format also needs to be converted into a single format for the ease of processing. I have adopted the TSO-structure in my decision support system, as an intermediate form of the messages that were originally written in a natural language. TSO is developed particularly for holding and exchanging emergency and disaster related situational information. It has a rich vocabulary (code) suitable for encoding situational information. Each TSO holds information using XML (Extensible Markup Language) which makes the construction of TSO easy.

It should be noted that although the Twitter API returns data in XML (or JSON) format, the actual message contained in a tweet is written in a plain natural language as shown in Section 3.1.1 and Appendix A.1. The structure of the tweet is irrelevant to the semantics of the message, i.e. the structure of the tweet is not designed to express its meaning, in any way. Therefore,

I use the TSO format to ensure that data is encoded in a single suitable format and it allows coding of natural language messages, which facilitates the automated analysis of the messages. Thus, TSO will make it easy to determine whether the information contained in a message is in conflict with that in other messages (the limited vocabulary makes consistency analysis more tractable). If each message was not converted into a TSO, it would not be possible to perform the analysis described in Section 7.1 (and shown in Figure 7.2 and Figure 7.3) and the scoring of the messages (see Section 8.2 and Figure 8.10). Being an XML document, each TSO allows the machine to read (without knowing the semantics) certain information such as the type of incident, the actors involved in the incident, number of casualty, etc. If we knew a method of generating Alloy specifications automatically, then having each message in TSO format would have enabled us to write the corresponding Alloy specifications mechanically, instead of hand coding them.

I currently show the encoding into TSO as internal to the system architecture. The TSO may be created manually by trained staff or potentially by an automated system. A detailed discussion on the TSO structure and its construction is given in Chapter 6. However, whether TSO can be automatically generated using Natural Language Processing (NLP) is a topic for future research.

### 5.1.3 Scoring Function

The scoring function (tagged with number 3) constitutes the second half of the *Organisational Policy* as shown in Figure 5.1. In order to evaluate the veracity of a message, the scoring function essentially uses the information provenance and other quality factors of interest (for a given context or incident) and specifies how to assign scores to messages for each factor. The selection of factors will depend upon context, since relevance and reliability of the factors as a trust indicator may vary according to their vulnerability to compromise, the likelihood of a compromise (malicious or accidental) taking place, as well as the nature of the incident. For example, in the case of a natural disaster (e.g. landslide, earthquake, etc.), *Location of Informer* (i.e. location of the source of information) is an important factor for assessing the correctness and reliability of the given information. This is due to the fact that people located far from the incident location may not know the information very well and even if they do know, they are unlikely to know the information at first hand.

A history of reputation can be maintained for each of the informants who provided information, which may be useful in future events/incidents. An informant may receive a positive score (e.g. +1) for each piece of information, s/he provided, which was correct and a negative score (e.g. -1) for each piece of information, which was incorrect. However, since reputation of an agent builds over time and it (reputation) is measured from past experience/interaction with the same agent, *Reputation* may not be an appropriate factor for assessing the correctness and reliability of information in this type of incidents, unless people in a particular region experienced similar incidents several times before.

Since, it is likely that the freshest (the latest generated) piece of information provides some previously unknown/undiscovered information, freshness of information can play an important role in prioritising information, especially when some pieces of information are contradictory. A time-stamp indicates the time when a piece of information was generated or released. Thus the time-stamp helps to determine the

Freshness and timeliness of a piece of information. Freshness of information can be used in two steps.

Firstly, it allows determining whether a piece of information is timely or stale i.e. whether a piece of information refers to an incident that took place in the past and we have closed that case and are not collecting information about that incident any more.

Secondly, if a piece of information is not stale, then we need to find out whether it is contradictory to any existing information or not. In principle, if a piece of information is supplementary or complementary to the existing information, then it may receive the highest score, (say) 10 and the score of the previous messages will remain unchanged. But if it is contradictory, then the latest information will receive the highest score, (say) 10 and the score of older pieces of information will reduce according to the Organisational Policy. For example, the scores of older pieces of information may be reduced by 1 for every additional one or two hours of their age, counting from the current processing time (or from the time when the latest but contradictory information was generated).

According to the policy, the scoring function will assign a vector of scores to each piece of information, one score for each of the provenance factors in use. In other words, the overall score given to a piece of information is a vector constituted of multiple scores; each score assigned against each of its provenance factors. Here is an example of two vector scores (**a** and **b**) that may be assigned to two different messages:

$$\mathbf{a} = (l, f, c) = (7, 6, 7)$$

$$\mathbf{b} = (i, l, f, r, c) = (3, 7, 8, 5, 6)$$

where,  $i \equiv$  Identity,  $l \equiv$  Location,  $f \equiv$  Freshness,  $r \equiv$  Reputation, and  $c \equiv$  Corroboration

The Organisational Policy will also specify how large or small a score should be, when a score will be assigned to each factor in a particular situation. For example, in an incident where eye-witness accounts are of paramount interest, then the policy may heavily penalise the locations that are not in the immediate area of interest. Likewise, where identity is of paramount interest and a white list and a black list is maintained for the information sources, then the sources on the white list may be given higher scores than those that are not on that list and the sources on the black list may be heavily penalised, if considered at all. The scored messages will be stored in a database for consistency analysis and conflict resolution.



### 5.1.3.1 Exemplification of Scoring Function

I have given an example of how a message can be scored according to a simple and contrived policy, which uses three of the trust factors, Location, Freshness and Reputation. How the location-score can vary according to the proximity of the informer to the incident location is defined in Table 5.1 and how the score varies depending on the age of information is set out in Table 5.2.

Table 5.1: Scoring Policy for Location

Proximity (Mile)	Score
0.5 or less	10
1 or less	9
2 or less	8
3 or less	7
4 or less	6
5 or less	5
7 or less	4
9 or less	3
10 or less	2
10 or more	1

Note: Proximity indicates the distance between the Location of Incident and the Location of Informant.

Table 5.2: Scoring Policy for Freshness or Timeliness

Age of Information (Hour)	Score
2 or less	10
4 or less	9
6 or less	8
8 or less	7
10 or less	6
14 or less	5
18 or less	4
24 or less	3
36 or less	2
48 or less	1

Note: The age of information is calculated from the current processing time i.e. it refers to the time elapsed since information was generated/released. The time when the information was generated/released is acquired from the timestamp generated by a third party e.g. Twitter, Facebook, and Mobile Networks.

Suppose an incident has taken place in Place-X on 30/06/2013 at 13:20 and two messages have been received that are reporting about the incident.

- Message-1 was sent by Person-A from Place-Y on 30/06/2013 at 13.25.
- Message-2 was sent by Person-B from Place-Z on 30/06/2013 at 16:00.
- Distance between Place-X and Place-Y is 0.5 mile.  
(i.e. the distance between the incident-location and the location of Person-A, when s/he sent the message, is 0.5 mile. )

Table 5.3: Scoring Policy for Reputation

InformerID	Score	Note
Person-B	6	Person-B provided six pieces of information in the past and all of them were correct.
abc@yahoo.com	4	abc@yahoo.com provided four correct pieces of information in the past.
xyz@yahoo.com	-2	xyz@yahoo.com provided four pieces of information in the past but three of them were wrong and only one was correct. Hence, $-3 + 1 = -2$

- Distance between Place-X and Place-Z is 4 miles.  
(i.e. the distance between the incident-location and the location of Person-B, when s/he sent the message, is 4 miles.)
- Person-A does not have a reputation history i.e. the score is 0.
- Person-B has a reputation score 6 (i.e. Person-B correctly provided six pieces of information in the past).

Based on the information listed above and the scoring policy set out in Table 5.1 and Table 5.2, the vector score  $(l, f, r)$  received by Message-1 and Message-2 will be  $(10, 10, 0)$  and  $(6, 9, 6)$  respectively.

How the overall total score of a World View can be calculated by combining individual vector scores of each message is discussed in Section 5.1.5.

#### 5.1.4 Consistency Analysis and Conflict Resolution

Decision making always involves reasoning and consistency analysis. Making reasonable decisions is unthinkable without logical reasoning. Since the objective of the decision support system is to construct one or more possible pictures (views) of a situation immediately after an incident to which the evidence collected from the messages pertains, it is imperative that a consistency analysis is carried out to ensure that the messages used to construct a view of a situation are consistent with the facts and physics of the world and are not contradictory to other messages in the same view. The decision support system that I have designed includes a reasoning unit (tagged 5 in Figure 5.1). The scored messages that are stored in the

database will be analysed to cluster the consistent messages together. Inconsistent or conflicting messages will be kept in different clusters. Thus, multiple clusters will be created with all messages in a cluster consistent and coherent. Hence, each cluster will represent a different view of the possible situation which is referred to as a *World View*. It should be noted that some of the messages may be included in multiple world views. Intuitively, each cluster consists of a maximal consistent subset of messages. Consistency analysis will be discussed in detail in Chapter 7.

### 5.1.5 Combiner Function: Calculating an Overall Total Score of a World View

The Combiner Function (tagged 7 in Figure 5.1) calculates the overall total score of a World View by combining individual scores assigned to each factor of each message by the scoring function.

Once a score is assigned for each provenance factor to all individual messages, the individual scores could simply be added to get an overall score of a world view like how eBay calculates their members' Feedback (reputation) score [39]. Calculating an overall aggregate score simply by adding individual scores for each factor means, an equal importance is given to all factors. However, all provenance and information quality factors are not equally important or applicable in different situations, as has been illustrated in Section 5.1.3. In some cases, some of the provenance factors are more relevant (and should get more priority) than others. Therefore, an overall score derived simply by adding the scores for individual factors for all messages is unlikely to produce a reliable score for the world views. The world view generated this way, with the highest overall score may always be far from what the evidence (from the messages) suggests. Hence, it is necessary to devise a method of deriving a composite trustworthiness-score from the raw scores that will produce an appropriate overall score for the world views, which will reflect their true reliability/trustworthiness.

The method of calculating an aggregate score varies depending on the requirement or context. For example, let us consider a university entry test scenario. Suppose, some students have received very good marks in Maths and Physics but a low mark in Biology, while some other students have received a high score in Biology but they are not so good in Maths or Physics. However, they have all received the same total score. Due to the selection criteria, the medicine department may calculate the aggregate score by somehow boosting the score in Biology, so that those who are good in Biology get priority in admission. On the other hand,

the Computer Science department may calculate the aggregate score by somehow boosting the score in Maths and Physics, so that those who have higher score in Maths and Physics get an overall higher score although everyone's total score (from their A level exam, for example) is same (as all subjects were treated equally). Many examination authorities use similar techniques to boost scores from some of the subjects or test when they work out a composite score from individual test scores [58]. Likewise, we need to find out which factors are relevant and more appropriate as a trust indicator in which context before we decide how the individual scores will be combined to calculate overall aggregate score for a world view.

Location of Informer is possibly the most relevant and/or important provenance-related factor in most (if not all) incidents, as the people who were at the scene of an incident and those who were close by are likely to have a better information about the incident. Therefore, it is desirable that the messages that received a higher score for the location factor have a bigger impact on the overall score of the world views. Hence, the values of such factors should be accentuated or amplified. In some cases, it may also be desirable to reduce the effect of some factors on the overall score of the world views. For example, the provenance factors that are more subjective may be less important in some cases and it may be too risky to depend on the subjective factors in some other cases. Some information (e.g. Corroboration, Freshness/Timeliness, Location of Informer, etc.) may be collected easily, securely and more reliably for which the user may have more confidence on the values of these factors than on those of others. Some information may also be more accurate than others. For example, accuracy of Freshness/Timeliness and Corroboration is likely to be relatively high (as they are easily computable), while the accuracy of Location is always likely to be relatively low, unless the location is detected automatically and reliably rather than depending on reporter's claimed location (even though it is apparently from GPS). Therefore, it may be important to reduce the impact of some of those factors or scores. However, attenuating the location score for any reason may be detrimental as location is a very significant factor in most cases.

In order to decide how the combiner function will accomplish the accentuation or attenuation of the scores, we will study the properties of different types of mean or average: Arithmetic Mean, Geometric Mean, Harmonic Mean and Quadratic Mean or Root Mean Square (RMS) of a set of numbers before presenting the combiner function.

### 5.1.5.1 Comparative Analysis of Different Average Functions

In this section, we will explore the properties of different average functions such as Mean (or Arithmetic Mean), Geometric Mean, Harmonic Mean and Quadratic Mean or Root Mean Square (RMS). Here is how different average values of numbers (e.g.  $x$ ,  $y$  and  $z$ ) are calculated :

- Mean = (Sum of all Elements in a Set)  $\div$  (Number of Elements in the Set)  
e.g.  $\frac{x + y + z}{3}$
- Geometric Mean = n-th root of the product of n numbers e.g.  $(x \times y \times z)^{1/3}$
- Harmonic Mean = (Number of Elements in a Set)  $\div$  Sum of Reciprocal of each Element e.g.  $\frac{3}{(\frac{1}{x} + \frac{1}{y} + \frac{1}{z})}$
- Root Mean Square = Square-root of the Average of the Square of each Element  
e.g.  $\sqrt{\frac{(x^2 + y^2 + z^2)}{3}}$

Due to the nature of the mathematical operations performed in each average function, they all produce different results for the same set of values. I have given examples of how the resulting values of different averages vary from each other. Consider the data in Table 5.4.

Table 5.4: Effect of Different Average Functions (a)

Data Series 1				Data Series 2				Data Series 3			
5				5				1			
5				4				3			
5				5				6			
5				4				1			
5				5				1			
5				6				2			
5				4				9			
5				6				8			
5				4				9			
5				7				10			
Mean	G.Mean	H.Mean	RMS	Mean	G.Mean	H.Mean	RMS	Mean	G.Mean	H.Mean	RMS
5	5	5	5	5	4.91	4.82	5.10	5	3.44	2.25	6.15

All values in the Data Series 1 are equal (i.e. its Standard Deviation is zero). Hence, their Arithmetic Mean, Geometric Mean and RMS values are all same, which

is 5. Although, all three series of numbers have the same average value, which is 5, the numbers in Data Series 2 have a low standard deviation, whereas the numbers in Data Series 3 have a high standard deviation. As a result, Geometric Mean and Harmonic Mean of the Data Series 2 (4.91 and 4.82 respectively) are greater than those of Data Series 3 (3.44 and 2.25). This indicates that while the Arithmetic Mean allows us to nullify the effect of having a low score by having another equally high score (and vice versa), Geometric and Harmonic mean penalises for having any low score in the series. In fact, Harmonic Mean is tougher than Geometric Mean in penalising for having a low score. This property of Harmonic Mean (and Geometric Mean) is even more evident in Table 5.5 and the corresponding graph in Figure 5.2.

Table 5.5: Effect of Different Average Functions (b)

	Series 1	Series 2	Series 3	Series 4	Series 5
	5	0.1	0.1	0.1	0.1
	5	5	5	3	3
	5	8	5	6	6
	5	7	4	1	1
	5	8	5	1	1
	5	9	6	2	2
	5	7	4	9	2
	5	9	6	8	1
	5	7	4	9	2
	5	10	7	10	3
<b>A. Mean</b>	5	7.01	4.61	4.91	2.11
<b>R.M.S.</b>	5	7.50	4.94	6.14	2.63
<b>G. Mean</b>	5	4.95	3.39	2.73	1.46
<b>H. Mean</b>	5	0.89	0.85	0.74	0.65

All values except one in the third column (WorldView 2) of Table 5.5 are very high. Therefore, the Arithmetic mean of the values is 7.01 whereas the Harmonic Mean is only 0.89 while Geometric mean is 4.95. If we compare the values of Harmonic Mean in Table 5.5, we can see that having large values has almost no effect on the overall score if there is only one very small value. On the other hand, Root Mean Square has the opposite effect. It rewards more for having the large values than penalising for having small values. This is why WorldView 4 (column 5) has a much bigger RMS value (6.14) than WorldView 3 (4.95), although their Arithmetic Mean is not very different. It is important to note that the RMS value is always greater than Arithmetic Mean while Geometric and Harmonic mean are always less than Arithmetic Mean.

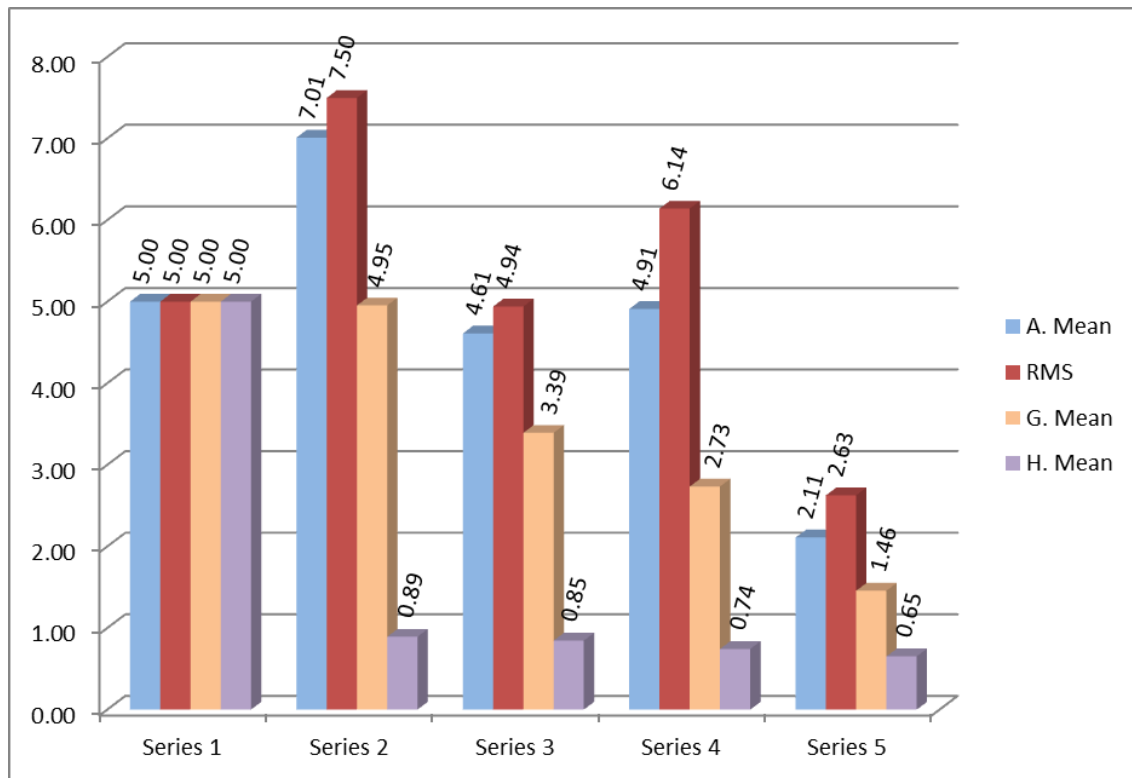


Figure 5.2: Effect of Mean, Geometric Mean, Harmonic Mean and Root Mean Square

In the light of the above discussion, we can justifiably use the RMS function for the values of those factors that we want to amplify and the Geometric Mean for those factors that we want to attenuate. However, this may raise a question that why Geometric Mean has been used instead of Harmonic Mean to attenuate values. The reasons are twofold:

1. Although the Geometric Mean reduces the effect of the large values when a data series contains small values, it does not mask large values as much as the Harmonic Mean does.
2. If a data series contains a zero then the Geometric Mean of the series becomes zero regardless how big the other values in the series are. However, the Harmonic Mean is even worse in handling a zero because it requires computing the reciprocal of the number (i.e.  $1/0$ ). This causes a computer to crash.

However, it may be necessary to avoid using Geometric Mean at times because Geometric Mean cannot handle negative numbers. In those cases, Harmonic Mean

may be an alternative option. Therefore, it has been suggested in [102] to allow the end user to choose the appropriate mathematical functions that suit their need.

#### 5.1.5.2 Combiner Function

The composite score of a world view is calculated by combining individual scores of different trust factors from all messages included in that world view, as illustrated in Table 5.6 with some example scores. How the scores are combined to produce the overall score for each world view is explained below:

1. Calculate the total score of the messages for each factor separately in the following manner (in any order):
  - Calculate the Mean score (Arithmetic Mean) for each of these factors separately: Freshness of Information, Context, and Corroboration
  - Depending on the type of incident and the policy adopted, calculate
    - Arithmetic Mean or Root Mean Square (RMS) of the scores corresponding to the factors Identity of Informer and Reputation
    - Arithmetic Mean or Geometric Mean (G. Mean) of the scores corresponding to the factors Social Relation and Competence

Generally, the Arithmetic Mean will be used to work out the overall score for all of these factors. However, in some cases (e.g. sensitive and delicate situations), we may want to give more emphasis to the messages received from some highly trusted sources with good reputation when we are certain about their identity (which may have been listed on a highly trusted list or not on a black list). In those cases, the RMS function will be used instead of Arithmetic Mean. It may be noted that Google Page-Rank algorithm uses similar techniques to work out the page-rank for a website. An external link from a website that has a higher page-rank carries more weight than a link from other sites that have lower page-rank, while other factors are constant [74]. Likewise, Geometric Mean will sometimes be used to attenuate the values instead of Arithmetic Mean.

- Take the Root Mean Square (RMS) of the scores for the factor Location of Informer



2. Calculate the Arithmetic Mean of the aggregate scores (found in step 1) for all factors. This average value (Arithmetic Mean) is the overall score of the world view.

Table 5.6: Calculating Composite Score

MsgId	Id	Loc	Fresh	Rep	Contx	Rel	Corro	Comp
1	6	7	8	6	0	0	3	4
2	4	6	4	5	2	1	6	4
3	3	7	5	0	0	0	3	5
4	5	8	7	5	2	1	7	6
	RMS/ Mean	RMS	Mean	RMS/ Mean	Mean	Mean/ G.Mean	Mean	Mean/ G.Mean
Overall Aggregate Score	=	A. Mean of the Aggregate Scores for each Factor i.e. Arithmetic Mean of the Results found in the previ- ous row						
MsgId $\equiv$ Message Id, Id $\equiv$ Identity of Informer, Loc $\equiv$ Location of Informer, Fresh $\equiv$ Freshness of Information, Rep $\equiv$ Reputation					Contx $\equiv$ Context/situation Inter- est & Ethics, Rel $\equiv$ Social Relation, Corro $\equiv$ Corroboration, Comp $\equiv$ Competence,			

The reason for using Geometric Mean and Quadratic Mean (Root Mean Square) has already been explained in Section 5.1.5.1. It has also been noted that Geometric Mean makes an overall score zero, if there is a zero among the scores although the rest of the scores are very high. However, it may not be very practical that (due to the Geometric Mean,) the scores of all messages for a specific factor gets entirely masked just because only one of the messages in the world view has a zero score for that factor. Therefore, the user may want to replace all zeros with a very small value (e.g. 0.01 or 0.001) so that the world view receives a reasonably minimised value instead of a zero.

Suppose a world view contains four messages and Table 5.6 shows the scores assigned to those (four) messages corresponding to different trust factors using the scoring function described in Section 5.1.3. Table 5.7 shows the overall total score of the world view following the procedure described above, assuming that this world view is associated with an event (incident), which is not very sensitive or delicate. (Hence, an arithmetic mean is be used to work out the overall score for each of the trust factors.)

Table 5.7: Calculating the Overall Total Score of a World View

MsgId	Id	Loc	Fresh	Rep	Contx	Rel	Corro	Comp
1	6	7	8	6	0	0	3	4
2	4	6	4	5	2	1	6	4
3	3	7	5	0	0	0	3	5
4	5	8	7	5	2	1	7	6
<b>Averages =</b>	<b>4.50</b>	<b>7.04</b>	<b>6.00</b>	<b>4.00</b>	<b>1.00</b>	<b>0.50</b>	<b>4.75</b>	<b>4.75</b>
<b>Overall Aggregate Score = 4.07</b>								

If the decision maker, for any reason, wants to give more priority to the location (3 times more, for example), s/he will multiply the corresponding scores of the location factor before calculating the overall score. In that case, the overall score of the world view (calculated from the values listed in Table 5.6 and Table 5.7) will be 5.83, as shown in Table 5.8. Chapter 8 demonstrates the construction and scoring of the world views, which make the whole process clearer.

Table 5.8: Change in Overall Total Score of a World View due to Decision Making Policy

MsgId	Id	Loc	Fresh	Rep	Contx	Rel	Corro	Comp
1	6	$7 \times 3$	8	6	0	0	3	4
2	4	$6 \times 3$	4	5	2	1	6	4
3	3	$7 \times 3$	5	0	0	0	3	5
4	5	$8 \times 3$	7	5	2	1	7	6
<b>Averages =</b>	<b>4.50</b>	<b>21.12</b>	<b>6.00</b>	<b>4.00</b>	<b>1.00</b>	<b>0.50</b>	<b>4.75</b>	<b>4.75</b>
<b>Overall Aggregate Score = 5.83</b>								

### 5.1.6 Decision Making Policy

The overall score of a world view is calculated by giving different weights to different factors depending on the context and according to their relevance, reliability and importance. However, it has been discussed in Chapter 4 that in addition to other factors, people's disposition to trust and propensity to taking a risk affect their decision. Therefore, the actual tool user, the decision maker, may want to change

some of the weights given to different provenance and quality factors. This is how the *Decision Making Policy* comes into effect.

Once the combiner function calculates the overall score of the world views and an ordered list of the world views is presented, the decision maker will then be able to change the ranking of world views according to their own policy, which is referred to as the *Decision Making Policy*. This policy will allow the decision maker to amplify or attenuate one or more provenance factors that may appear to be more or less important in a particular situation, in their opinion, by increasing or decreasing the weight given to the corresponding score across all messages in all world views. This feature may be implemented in part via a tool akin to a graphic equaliser, where the user can effectively turn up or turn down the relative importance of a factor against others. This will change the ranking of the world views of a possible situation. However, it should be noted that incorporating the decision maker's choice or view will not always make the world views better<sup>1</sup>. It may even take the world views further away from what has been really suggested in the messages. Therefore, the scope of the decision making policy is restricted to changing only the weights of the provenance and trust factors. Decision makers cannot change any initial score assigned to an individual message corresponding to the provenance and trust factors; they only increase or decrease the weight of certain factors, which only affects the overall scores of the collections of messages.

The policy maker may also set some threshold values for different factors when s/he does not want to consider messages with a score above/below the threshold value for the specified provenance factors. For example, if the decision maker wants to receive messages only from the affected area, then s/he may set a threshold value of 6 (say) for the provenance factor Location. Hence, any message that receives a score less than 6 for location will be ignored.

## 5.2 Summary

In order to reduce uncertainty in the wake of crisis and emergency and to help making an informed decision by assessing the reliability and trustworthiness of in-

---

<sup>1</sup>The word 'better' has both a subjective and an objective interpretation. In the true (objective) sense, a better world view gives a more precise picture of the situation (i.e. what has really happened on the ground). However, it is never possible to know which one is a better world view until the truth is discovered after the crisis/uncertainty is over. However, a Decision Maker may choose a set of parameters to produce a world view, that s/he believes is "most likely" to reflect the reality (about which, s/he may, of course, be wrong.)

formation from various sources, a decision support system (DSS) has been presented in this chapter which is unique for its novel design. This DSS makes a novel use of open source information in order to generate one or more possible scenarios of an incident for helping the emergency responders understand a situation better. A general discussion of how different components of the system will operate has also been presented in this chapter. The decision support system collects messages (information), encodes them into TSO and scores each TSO according to the policy driven trust metric. The DSS then performs knowledge-based consistency analysis of the messages to construct an unordered set of possible world views each of which is internally consistent and calculates the aggregate score for each World View in order to present the world views in the order of their scores. This chapter also includes a broad discussion on how individual scores associated with different provenance factors should be combined to produce the overall reliability score of the world views. However, the operation procedure of some of the components of the DSS has not been detailed in this chapter. They are covered in the following chapters. Chapter 6 will describe the structure of TSO in detail, the changes required in the original structure of TSO, and how a complete schema for the TSO is constructed. Chapter 7 will detail the methodology of constructing the World Views through logic based automated reasoning and consistency checking. The implementation process of the entire decision support system will be explained in Chapter 8 and some experiment results using the DSS will be presented in Chapter 9, before the concluding chapter of the thesis.

## CHAPTER 6

# Structuring Data into Suitable Format

Information collected from different sources comes in a variety of different structures and formats, if they are not totally unstructured. This poses a challenge in utilising crowdsourced and heterogeneous information. To overcome this problem, it is imperative to convert all heterogeneous information into a single structure/-format as shown in the system architecture (step 2) of the decision support system (Figure 5.1) that will make the data processing and analysis tractable. In Chapter 5 (Section 5.1.2), it was mentioned that the TSO may be created manually by trained staff or potentially by an automated system. This chapter describes how the manual (hand coded) conversion of data into a single and uniform TSO-structure, in the context of the decision support system, is achieved and what has been done to achieve this with relative ease.

### 6.1 Tactical Situation Object (TSO)

OASIS (Open Advanced System for dISaster & emergency management) was a European Framework (FP6) Project focused on Emergency and Disaster Management [42]. While working with the aim to design a generic crisis management system, OASIS proposed a standard data format for exchanging information between systems during disaster and emergency management. The Tactical Situation Object (TSO) is an object containing language-independent situation information encoded in XML following the format designed and proposed by OASIS. The TSO can be used to describe an event or incident, the resources available or deployed in the operation after the incident, and the missions in progress. However, I have used TSO only to describe the information found in each message i.e. the TSO is an intermediate form of a tweet (a Twitter post), which allows coding of natural

language messages. Thus, the TSO facilitates the extraction of the semantics of messages. Hence, each TSO, in essence, is a message, which corresponds to a tweet.



Figure 6.1: A Tactical Situation Object (TSO) Created by the Emergency Services Call Centre [43]

### 6.1.1 Structure of a Tactical Situation Object (TSO)

A TSO has four top level elements/components: *<CONTEXT>*, *<EVENT>*, *<RESOURCE>*, and *<MISSION>*. Among them, *<CONTEXT>* and *<EVENT>* are compulsory and the rest are optional. (However, as TSO is an XML

```

<?xml version="1.0" encoding="UTF-8"?>
<TSO_2_0 xmlns="http://tacticalsituationobject.org/schemas/TSO/2_0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <CONTEXT>
    <ID>SDIS35_200711191727_033</ID>
    <MODE>ACTUAL</MODE>
    <MSGTYPE>ALERT</MSGTYPE>
    <CREATION>2007-11-19T17:27:00.0Z</CREATION>
    <URGENCY>URGENT</URGENCY>
    <ORIGIN>
      <ORG_ID>FR_SDIS35</ORG_ID>
    </ORIGIN>
  </CONTEXT>
  <EVENT>
    <ID>SDIS35_200711191727_EV033</ID>
    <DECL_DATIME>2007-11-19T17:24:00.0Z</DECL_DATIME>
    <REFERENCE>
      <ORG_ID>FR_112_DEP35</ORG_ID>
      <OTHER_EVENT_ID>CC112_200711191720_EV03</OTHER_EVENT_ID>
    </REFERENCE>
    <EGEO>
      <TYPE>GEN/INCGRD</TYPE>
      <POSITION>
        <LOC_ID>BETTON_ACC</LOC_ID>
        <TYPE>POINT</TYPE>
        <COORD>
          <LAT>48.18</LAT>
          <LON>-1.63</LON>
        </COORD>
      </POSITION>
    </EGEO>
  </EVENT>
  <RESOURCE>
    <RTYPE>
      <CLASS>/MAT/VEH/RAODVEH/FRFGTN/RSC</CLASS>
    </RTYPE>
    <ID>VSAB_BET01</ID>
    <RGEO>
      <DATIME>2007-11-19T17:32:00.0Z</DATIME>
      <TYPE>INC</TYPE>
      <POSITION><LOC_ID>BETTON_ACC</LOC_ID></POSITION>
    </RGEO>
    <STATUS>IN_USE/MOBILE</STATUS>
  </RESOURCE>
  <RESOURCE>
    <RTYPE>
      <CLASS>/MAT/VEH/RAODVEH/FRFGTN/FRF</CLASS>
    </RTYPE>
    ...
  </RESOURCE>
  <MISSION>
    <TYPE>/SAV/RTA</TYPE>
    <STATUS>IPR</STATUS>
    <RESOURCE_ID>VSAB_BET01</RESOURCE_ID>
  </MISSION>
  <MISSION>
    ...
  </MISSION>
</TSO_2_0>

```

Unique TSO file identifier (for the Fire services node)

**Important:** this is the unique identifier of the originating node (the fire service node)

The event identifier used in the Fire service node.

**Important:** the event is the same than the event declared in the 112 Call Centre

Location is still the same, no casualty assessment

A rescue engine is sent from the fire station to the incident place, with an estimated time of arrival (ETA)

Another fire engine is sent from the fire station to the incident place (this section is not detailed here)

2 missions are defined for the 2 resources

Figure 6.2: A Tactical Situation Object (TSO) Created by Fire Service in Response to the TSO in Figure 6.1 [43]

Object, all of these top level elements come under the root element of the TSO e.g. `<TSO_2_0>`). `CONTEXT` contains information such as whether the message describes an actual event or, it is an exercise or test event. `CONTEXT` also includes identification information (e.g. Identification of each Message, Identification of its Originator, the time of its creation, etc.) and indicates the purpose of the message e.g. Alert, Acknowledgement, Update, etc. `EVENT` contains a description of the event including the nature of incident, who/what is involved in the incident, number of casualties, environment of the incident place, etc. `RESOURCE` describes all information relating the deployment of resources, such as the Police, ambulance, fire engine, etc. while `MISSION` describes the mission(s)/task(s) assigned to each resource or agent, their progress, missions foreseen, etc. Hence, when a TSO only contains the initial report of an incident, naturally this TSO has only two components, `CONTEXT` and `EVENT` as shown in Figure 6.1 [43]. The TSO shown in Figure 6.1 contains information only about the incident, a collision between a truck and a train at a level crossing, received by the operator at the emergency services call centre. The TSO shown in Figure 6.2 contains additional information from the Fire Services as they respond to the traffic incident described in the initial TSO (shown in Figure 6.1).

However, in the context of my Decision Support System, I only need those elements of TSO that are essential to describe a (likely) situation according to a message (a tweet, for example). Hence, I have only used the elements, `<CONTEXT>` and `<EVENT>`, and some of their sub-elements. The example below gives a clear mapping of a message into a TSO and demonstrates which of the elements of a TSO are useful in our case.

Suppose, we have received the following Twitter message:

*“Accident: I’ve seen a collision between a Train and a Truck on the level crossing near Coventry Station: 10 Casualties, all in Critical Condition & require Emergency Care”*

The TSO, which corresponds to this message, is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<TSO_2_0>
  <CONTEXT>
    <ID>COVENTRY_20110305_172500</ID> <!-- A unique ID is assigned to every TSO -->
    <MODE>ACTUAL</MODE> <!-- Real Incident -->
    <MSGTYPE>ALERT</MSGTYPE>
    <CREATION>2011-03-05T17:25:00.0Z</CREATION>
```



```

<URGENCY>URGENT</URGENCY>
<ORIGIN>
  <ORG_ID>Call_Centre.01</ORG_ID>
  <!-- Next two elements, SRC.ID & SRC.LOCATION, are not defined in the original TSO Schema
  or Data Dictionary -->
  <SRC_ID>Twitter_User.0001</SRC_ID>
  <SRC_LOCATION>52.404575,-1.514697 (Warwick Road, Coventry)</SRC_LOCATION>
  <!-- Location Associated with the tweet -->
</ORIGIN>
</CONTEXT>
<EVENT>
  <ID>Not_Applicable</ID> <!-- Although (Event) ID is a compulsory element for a TSO (by
  design), I have not used it as we don't know what the event is, in the first place. Hence, the value
  of the element is 'Not Applicable' -->
  <NAME>ACCIDENT TRAIN COVENTRY</NAME>
  <ETYPE>
    <CATEGORY>
      <TRP>COL</TRP> <!-- Transport Collision -->
    </CATEGORY>
    <ACTOR>
      <VEH>
        <TRN/> <!-- Train -->
        <TRK/> <!-- Truck -->
      </VEH>
    </ACTOR>
    <LOCTYPE> <!-- Incident Location is a Level Crossing -->
      <RAIL>TRK</RAIL>
      <ROAD/>
    </LOCTYPE>
  </ETYPE>
  <SOURCE>HUMOBS</SOURCE> <!-- HUMAN OBServation i.e. Eye-witness
  Account -->
  <DECL_DATIME>2011-03-05T17:24:00.0Z</DECL_DATIME>
  <OCC_DATIME>2011-03-05T17:24:00.0Z</OCC_DATIME>
  <CASUALTIES>
    <CONTEXT>PRELIM_STAT</CONTEXT>
    <TRIAGERED>10</TRIAGERED>
  </CASUALTIES>
  <EGEO>
    <TYPE>
      <GEN>INCGRD</GEN>
    </TYPE>
    <POSITION>
      <LOC_ID>TRN_ST_COVENTRY</LOC_ID>
      <NAME>Railway Station, Coventry, UK</NAME>
    </POSITION>
  </EGEO>
</EVENT>
</TSO.2.0>

```

## 6.2 Usability Issues of the TSO Data Dictionary

Since TSO is an XML object, it is necessary that there is a comprehensive schema to support or guide the production of TSO. However, the CEN Workshop Agreement [43, 44] only provides the message structure (similar to a document object model) of TSO (which also includes a normative schema) and a complementary data dictionary. The message structure defines the elements of the TSO, their data type and other data restrictions. While the message structure also defines some of the possible values of different elements, most sub-elements of a TSO and their possible values (codes) are listed in the other part of the TSO documentation, the data dictionary. Table 6.1, which is an excerpt from [43], shows how different elements of TSO (more specifically, some of the sub-elements e.g. ID, NAME, CATEGORY, ACTOR, etc. of the EVENT element of TSO) have been defined and the light-green shaded cells in the last column of the table refer to the data dictionary for the values or codes of some TSO elements.

Another table (Table 6.2), which has been adapted from the Data Dictionary [44], shows the sub-elements of event CATEGORY. Although, in theory, it will not be utterly impossible to create a TSO by manually looking up the codes or values for each of the elements of TSO from the data dictionary, it is, no doubt, an impractical method. Hence, it is essential to provide a complete schema for creating TSO with a reasonable effort. I have combined the data dictionary with the normative schema provided by OASIS in order to construct a comprehensive schema for the TSO. The normative TSO schema was only 1,337 lines long while the file size was 58.4KB, whereas the complete schema consists of 5,743 lines and the file size is 480KB.

## 6.3 Constructing TSO Schema from TSO Message Structure and its Data Dictionary

Since the only source of the TSO message structure and its data dictionary was the two PDF files published as CEN Workshop Agreements [43, 44], I did not have any choice but to extract the data by converting the files into Microsoft Word and Excel format respectively. The incomplete schema extracted from [43] was then saved as an XML Schema Definition (XSD) file, although I had manually corrected many errors made by the PDF converter. However, extracting the data tables from the data dictionary was even harder. Since it would not be possible to construct the full

Table 6.1: TSO Message Structure defines some of the Elements and Sub-Elements [43]

Element Name	Hierarchy	Type	Cardinality	Definition	Notes or Value Domain
ID	TSO_2_0 EVENT	string (maximum 40 characters)	REQUIRED, 1	Describes the identifier of the event for the creator of the TSO	This identifier shall be unique inside the node.
NAME	TSO_2_0 EVENT	string (maximum 40 characters)	OPTIONAL, 0 or 1	Provides a name for the event	It is the responsibility of the node to have unique event names.
MAIN_EVENT_ID	TSO_2_0 EVENT	string (maximum 40 characters)	OPTIONAL, 0 or 1	Provides a link to the main event	In this case, the current event is a sub-event of this main event
ETYPE	TSO_2_0 EVENT	group	OPTIONAL, 0 or 1	The type of the event, which is the collation of several facets	
CATEGORY	TSO_2_0 EVENT ETYPE	String (maximum 80 characters)	REQUIRED, [1..n]	The description of the scenario which leads to the event	For values see the TSO Code Definition. The complete list of values is in the data dictionary.
ACTOR	TSO_2_0 EVENT ETYPE	string (maximum 80 characters)	REQUIRED, [1..n]	Describes the type of the endangered object(s).	For values see the TSO Code Definition. The complete list of values is in the data dictionary
LOCTYPE	TSO_2_0 EVENT ETYPE	string (maximum 80 characters)	REQUIRED, [1..n]	Describes the type of the location where the event is taking place.	For values see the TSO Code Definition. The complete list of values is in the data dictionary
ENV	TSO_2_0 EVENT ETYPE	String (maximum 80 characters)	OPTIONAL, [0..n]	Describes the general environment (or context) of the event	For values see the TSO Code Definition. The complete list of values is in the data dictionary
SOURCE	TSO_2_0 EVENT	String (enumeration)	OPTIONAL, 0 or 1	Describes the origin of the declaration of the event	Possible values are: COMFOR (computer forecast), HUMDED (human deduction), HUMOBS (human observation), SENSOR (sensor observation).

schema manually, I wrote an application software for this purpose. However, I had to store the extracted data into an SQL Server database to enable the application to do the work. Storing this data into SQL Server database was not a straight forward process. First, I converted the source file into Microsoft Excel format and saved each table into a separate worksheet. This was done because Microsoft Access (a lightweight relational database application) can import each Excel worksheet as a

Table 6.2: Codes defined in TSO Data Dictionary (a) [44]

/EVENT/ETYPE/CATEGORY

	Higher levels	Code	Definition	Additional description
1		ASB	Anti social behaviour	Anti social behaviour
2		ASR	Assistance or rescue for person/animals	Assistance or rescue for person/animals
3		EXP	Explosion	Explosion
4		FIR	Fire	Fire
5		FLD	Flood	Flood
6		GND	Ground Event	Ground Event
7		HLT	Health	Health
8		POL	Pollution	Pollution
9		PSW	Public safety/welfare	Public safety/welfare
10		TRP	Transport	Transport
11	/ASB	ABV	Abandoned vehicle	Vehicle that has been left unattended
12	/ASR	ATM	Smoke/un-breathable atmosphere	Atmosphere that requires specialist breathing apparatus to sustain life
13	/ASR	HGT	Rescue from a height	Emergency rescue from a height above the normal reach of standard rescue equipment
14	/ASR	ICE	Rescue Ice	Rescue of casualty from, on or under ice
15	/ASR	MAR	Marooned	Person/s in a position or situation where self rescue is impossible
16	/ASR	SIL	Rescue silos/sand	Rescue operations within silos
17	/ASR	TRP	Trapped	Situation whereby a casualty cannot self rescue
18	/ASR	UDG	Rescue underground	Rescue operations below ground level
19	/ASR	WAT	Rescue Water	Rescue operations from water
20	/EXP	AER	Aerosols	A gaseous suspension of fine solid or liquid particles, packaged under pressure with a gaseous propellant for release as a spray of fine particles
21	/EXP	AMM	Ammunition	Explosives used in weapons such as bullet, shells, etc.

database (Access) table and SQL Server can import data from Access. SQL server cannot import data directly from Excel. Once the data importation and correction is complete, my application which is, in fact, an XML parser and generator, started building the complete schema. The core technique of developing the schema is explained below.

Notice the underlined text “/EVENT/ETYPE/CATEGORY” which is written just above the Table 6.2. The underlined text above the table says that all the elements listed in the table are sub-elements of ‘CATEGORY’, which itself is a sub-element of EVENT and ETYPE. We have learnt earlier in this chapter that ‘EVENT’ is one of the top-level elements of TSO that comes only under the root

Table 6.3: Codes defined in TSO Data Dictionary (b) [44]

/EVENT/ETYPE/ACTOR

	Higher levels	Code	Definition	Additional description
1	/PPL/CHD	BAB	Baby (under 12 months)	Baby (under 12 months)
2	/PPL/CHD	INF	INFANT (between 1 and 3 years)	Less than 3 years (reduced mobility and understanding)
3	/PPL/CHD	CHILD	Child: between 3 years and 10 years	A child is approximately between 3 and 10 years.
4	/PPL/GND	FML	Female	Female person
5	/PPL/GND	MAL	Male	Male person

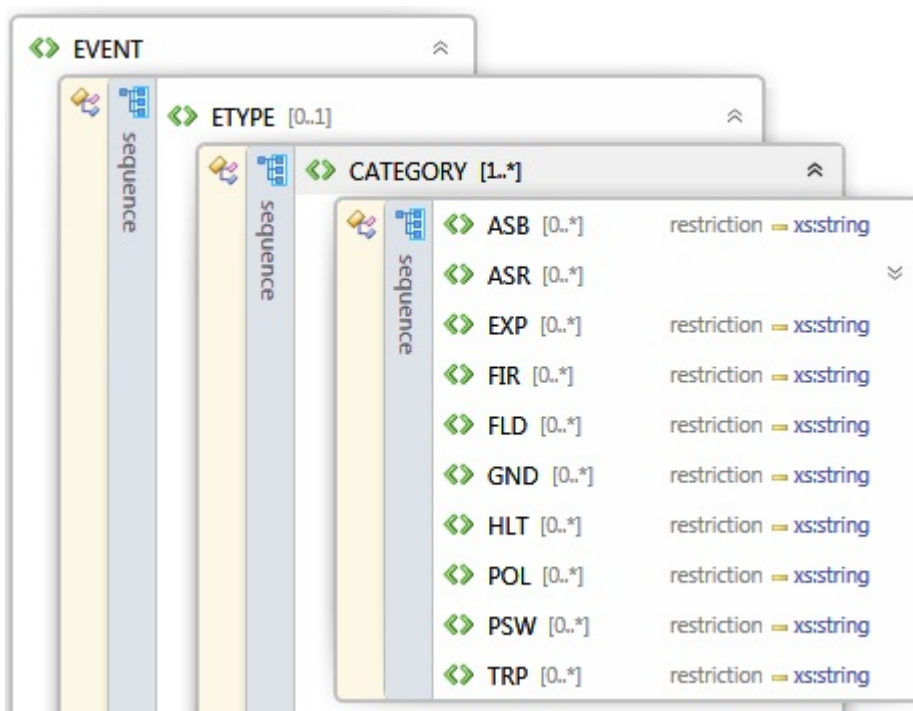


Figure 6.3: Hierarchy of some of the TSO Elements

element.

The first ten codes listed in the second column (**Code**) of the table do not have a corresponding value in the first column (**Higher levels**). This indicates that these ten elements are the direct child elements of **CATEGORY** as shown in Figure 6.3. For example, the hierarchical position of **ASB** will be **/EVENT/ETYPE/CATEGORY/ASB**.

The 11th row of the table contains **ASB** in the first column and **ABV** in the

second column. This means **ABV** is a direct child element of **ASB**. Therefore, the hierarchical position of **ABV** will be `/EVENT/ETYPE/CATEGORY/ASB/ABV`. Likewise, the codes in the rows from 12 to 19 are the direct child elements of **ASR**. Notice that the codes defined in Table 6.3 have two codes in the ‘Higherlevels’ column instead of one. Therefore, the parent elements of the first three codes in Table 6.3 is `/EVENT/ETYPE/ACTOR/PPL/CHD` and of the last two codes is `/EVENT/ETYPE/ACTOR/PPL/GND`. Thus, it is possible to identify the relative position of a TSO code from the data dictionary.

Before I explain how new elements from the data dictionary are added to the existing (incomplete) schema, it is important to know about different types of elements an XML file can have. An XML file can have two types of elements:

1. Complex Element: A complex element can have any number of child elements that can be both simple and complex.
2. Simple Element: A Simple element cannot have any child element but a value of a single data type e.g. string, integer, dateTime, etc. If a simple element accepts only a certain number of fixed values, then those values are specified as its ‘enumeration’ values.

### 6.3.1 Adding new Elements to the Existing Schema

Suppose, `CATEGORY (/EVENT/ETYPE/CATEGORY)` is defined in the initial schema as a simple-type element. However, Table 6.2 indicates that `CATEGORY` can have multiple sub-elements. Hence, we need to declare `CATEGORY` as a complex-type element before appending any child element to it. However, if an element that can have sub-elements was already defined as a complex element, then sub-elements can be appended without making any change to the element (parent to be).

*What would happen if the element `CATEGORY` (parent to be) was defined as an ‘enumeration’ value of another simple element?*

Since the application adds only one new element/code at a time, if `CATEGORY` was an ‘enumeration’ value of another simple element then following changes would have to be made:

1. Redefine the parent element of `CATEGORY` as a complex element.

2. Define CATEGORY as a simple element.
3. Add the new code from the data dictionary as an ‘enumeration’ value of CATEGORY instead of defining it as a child element.

This process continues to ensure each code listed in the data dictionary is added to the schema. However, any code that is already included in the schema is skipped to avoid duplication. The top level elements of the complete TSO schema are shown in Appendix D.

## 6.4 Modification and Extension of TSO

While constructing XML schema for Tactical Situation Object by combining the OASIS schema and their data dictionary, I have found some design problems in TSO Data Dictionary. One is related to naming an element and the other is related to specifying the type of an element.

### **Renaming and Redefining the Type of TSO elements**

One of the codes in the TSO data dictionary, TRN (EVENT/ETYPE/ACTOR/VEH/TRN, Train), can have multiple values including 3RL, DSL and HZD. So, these values could be added to the restriction (Enumeration Values) of the element, TRN, in the TSO schema as none of them (3RL, DSL, HZD, etc.) can have sub elements. However, while DSL, HZD, etc. are valid identifiers but XML compiler does not accept 3RL as a valid identifier as it starts with a number. The XML compiler gives the following error message: “Elements with the same name and in the same scope must have the same type”. Another code (element) in the TSO data dictionary, PPL (EVENT/ETYPE/ACTOR/PPL, People), can also have multiple values (sub-elements) such as 1 (one person), ADU (adult), CHD (child), etc. However, these values cannot be added as Enumeration Values because some of them (e.g. CHD) can have child elements whilst others cannot (e.g. 1). Therefore, I have no option but to define the elements like PPL as a Complex Type element in the TSO schema and its child elements as Simple Type elements. Thus, a portion of our TSO Schema will look like:

```

<ACTOR>
  <PPL>
    <1/>
    <CHD> BAB </CHD>
  </PPL>
</ACTOR>

```

which is similar to:

```

<Person>
  <Sadiq/>          <!-- 'Sadiq' should have been a value of another element-->
                    <!-- e.g. <Name>Sadiq</Name> -->
  <Address> University of Warwick </Address>
</Person>

```

However, this leads to two more problems:

- a) 1 is considered to be an element whilst it should be a value.
- b) < 1 > is not a valid XML element because XML element names cannot start with a number or punctuation character.

Table 6.4: Some TSO Codes Need Renaming

Parent Elements	Original Code	Proposed Code	Definition
EVENT/ETYPE/ ACTOR/PPL	1	ONE	A single Person
EVENT/ETYPE/ ACTOR/VEH/TRN	3RL	RL3	Motive power: Electricity 3rd rail
EVENT/ETYPE/ LOCTYPE/ROAD	1RD	RD1WAY	One-way Road
RESOURCE/RTYPE/ CLASS/MAT/ CM/MEDICN	1STAID	AID1ST	First Aid

Therefore, some of the TSO codes found in the data dictionary have been renamed, as shown in Table 6.4, in order to produce a valid XML document. Another problem found in the TSO data dictionary is that the same code (e.g. HUM) is used to refer to two different things in different context:



- HUM: Humid conditions [/EVENT/EGEO/WEATHER/HUM]
- HUM: Human resource [/RESOURCE/RTYPE/CLASS/HUM]

### Elements and Enumeration Values are Made Optional

*Empty String as an Enumeration Value:* At times, it becomes necessary to have a TSO element with no content in it. For example, a train was involved in an accident but we do not have any more information about the train, whether it was a passenger train or it was carrying any hazardous load, etc. In that case, we need to have the TRN element with an empty content e.g.

```

<ACTOR>
    <VEH>
        <TRN/>
    </VEH>
</ACTOR>

```

However, the current TSO schema does not allow the TRN element to be empty. It has to have either of these enumeration values:

```

<xs:enumeration value="RL3"/>
<xs:enumeration value="DSL"/>
<xs:enumeration value="HZD"/>
<xs:enumeration value="LOC"/>
<xs:enumeration value="NHZ"/>
<xs:enumeration value="NUK"/>
<xs:enumeration value="OVH"/>
<xs:enumeration value="PAS"/>
<xs:enumeration value="REF"/>
<xs:enumeration value="STM"/>
<xs:enumeration value="TRM"/>
<xs:enumeration value="UDG"/>
<xs:enumeration value="UND"/>
<xs:enumeration value="VIP"/>

```

```
<xs:enumeration value="VLT" />
```

Therefore, I have added another enumeration value to it, `<xs:enumeration value="" />`, so that the TRN element can be used with an empty content (`<TRN/>`) instead of something like `<TRN>PAS</TRN>`. Making the value of an element optional is important because some information may be unavailable at times. For example, a reporter may simply say a train was involved in an accident without giving any more details such as a passenger train or a goods wagon. However, an empty enumeration value has been added only to those elements that have received one or more enumeration values from the data dictionary. In other words, no empty enumeration value has been added to the elements (with enumeration values) that already existed in the original TSO schema (provided by OASIS) and they have not taken any enumeration value from the data dictionary.

#### *Elements made Optional:*

Since, the TSO data dictionary does not say anything about whether an element is optional or compulsory for a TSO and how many times an element may appear in a TSO, all elements added to the TSO schema from data dictionary are made optional. These optional elements may also appear any number of times in a TSO. However, the elements that already existed in the original schema have remained as original.

### **Extension of TSO**

While the original TSO structure is designed to capture the information as to which organisation has created the TSO and who was the operator, it cannot capture who has reported the information and the location of the reporter. In order to collect the identification and location of informer, I have added two new elements, `<SRC_ID>` and `<SRC_LOCATION>` respectively, to the TSO structure (e.g. `<SRC_ID>ABC@YAHOO.COM</SRC_ID>`, `<SRC_LOCATION>CV4 7AL</SRC_LOCATION>`).

## 6.5 Tactical Situation Object (TSO) Encoding

With the help of the complete and independent TSO schema that I have generated, any standard XML editor can be used to create TSO manually. Chapter 8 demonstrates the encoding of a message into TSO and shows how the complete and independent schema helps the process. Although, the encoding of messages into TSO has to be performed with hand, the (complete) schema has made the task similar to filling in a web form. However, it is expected that a large volume of messages will be received during a crisis and emergency and those messages need to be processed reasonably quickly. Therefore, the encoding of messages into TSO may have to be performed by an appropriate number of trained staff (who may be located in different geographic locations in order to save cost). However, despite the provision of a complete schema and the available tools for TSO construction, manual encoding of TSO makes the whole system fall short of complete automation. This is due to the fact that the biggest hurdle in encoding the messages into TSO, completely automatically, is apprehension of the messages that are written in a natural language. Investigating the possibility of generating TSO automatically using natural language processing is beyond the scope of this thesis and this is a topic for future research.

## 6.6 Summary

In order to ensure that all data collected from various sources are encoded into a single format that will make automated data analysis easier, a standard data format has been adapted which was originally proposed for exchanging information between systems during disaster and emergency management. The structure of the adapted data format which is referred to as Tactical Situation Object (TSO) has been described in this chapter. The construction of a complete schema for the XML based TSO and its extension will make it easy for us to encode messages into TSO. Since a limited vocabulary makes consistency analysis tractable, the TSO will also enable us to easily determine where the information contained in messages is in conflict with that in other messages. It has been stipulated in the description of the system framework (in Chapter 5) that the messages need to be encoded into TSO before assigning the scores against the provenance and trust factors, and performing the consistency analysis. Since, TSO has a finite number of codes (i.e.

limited vocabulary) for encoding messages in the context of disaster and emergency management, any message that is out of the scope of TSO (i.e. cannot be encoded into TSO) gets discarded automatically. Thus, TSO format ensures that the system needs to deal with only a finite number of messages. How the consistency analysis of the messages is performed will be described in Chapter 7. Although the fully automatic construction of TSO could not be achieved, how messages can be easily encoded into TSO using any standard XML editor (due to the construction of the complete TSO schema) will be demonstrated in Chapter 8.

## CHAPTER 7

# Methodology of Constructing World Views

Decision making always involves reasoning and consistency analysis. Making a rational decision is unthinkable without reasoning and consistency analysis. Therefore, it is important for both human and machine that they perform logical reasoning before making a decision (or supporting decision making). One of the core components of the Decision Support System that I have designed is the Consistency Analysis and Conflict Resolution Unit (Unit 5 on the system architecture shown in Figure 7.1). The Consistency Analysis and Conflict Resolution Unit carries out consistency checking in order to resolve contradictions between messages and to construct world views, each of which contain coherent messages. However, before checking the consistency (or finding a contradiction) between two or more messages, it is necessary to ensure that those messages are relevant to each other (not discussing about disparate subjects). Two or more messages can be truly consistent or contradictory with each other, only when they are related or relevant to each other. The following messages highlight this phenomenon.

Consistent messages:

- “The Internet may be very unsafe for children”
- “Parental controls and other security software are likely to improve their safety on the Internet”

Contradictory messages:

- “The law and order situation in the city has recently improved”
- “Due to growing unemployment, all sorts of crime has recently increased in the city”

Unrelated and disparate messages cannot contradict each other. Therefore, a logical system will not find any contradiction between irrelevant messages. Consider the following messages as an example:

- “A tourist coach has crashed in Oxford”
- “Signal failure in London underground”
- “There are lot of people in Victoria Park”

These three messages are not contradictory but totally irrelevant to each other. Hence, there is no point in checking the consistency of such irrelevant messages. In order to reduce the processing overhead, irrelevant messages need to be separated before the logical consistency checking is performed. The scored messages, therefore, need to be stored in the database in the (logical or conceptual) order shown in an expanded version of the system architecture in Figure 7.1 (step 4b). Since TSO-structure has been adopted in my decision support system and TSO specifies the date, time, location and type of an incident and the actors involved in that incident, irrelevant messages can be separated with a simple query (using XQuery and/or SQL). However, in a larger system which can be used in real-life cases, tools like Apache Solr, an open-source search engine, and Carrot2, an open-source search results clustering engine, may be used in the future to cluster the relevant messages together before TSOs are created.

In this chapter, I explain in detail how the self-consistent world views are created through logic based consistency analysis in detail. In brief, I have treated the consistency analysis and finding coherent world views as an application of constraint satisfaction and model finding. Hence, I have used Alloy, a model specification language, and its analyser (Alloy Analyzer<sup>1</sup> [sic]) to perform this task automatically. However, before describing the logical consistency checking system, I will describe a different method of constructing world views which will motivate the adoption of logic based approach.

---

<sup>1</sup><http://alloy.mit.edu/alloy/>

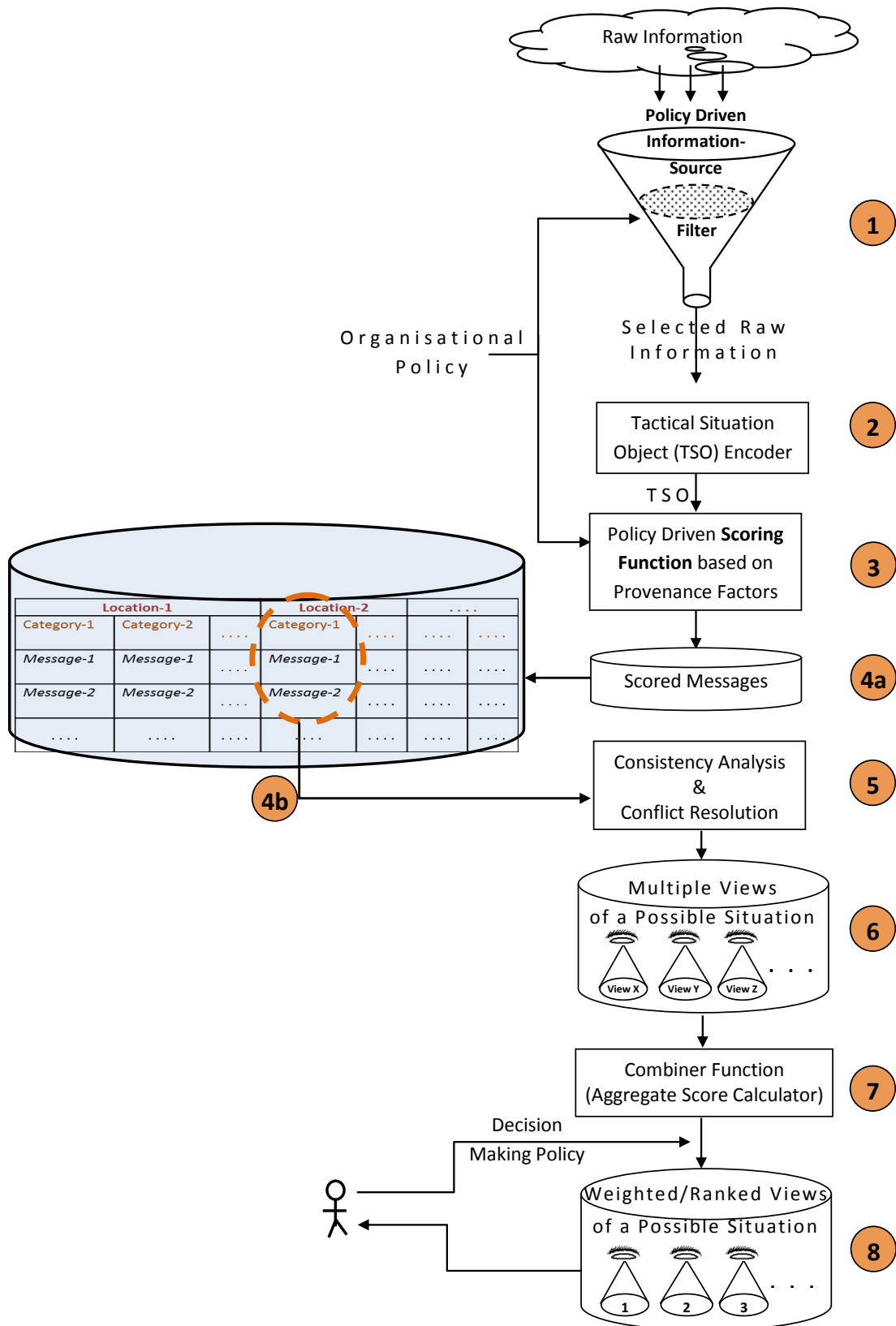


Figure 7.1: Data Flow in the System Architecture: An Extended View

## 7.1 Clustering TSO based on Event Location, Event Type, Actors and Sentiment/Positivity

Since the objective of the decision support system is to construct possible world views of a situation at a particular place from the collected messages, all messages associated with that specific place or location should be grouped and considered together. However, multiple incidents may concurrently take place at a location and those incidents are more likely to be of multiple categories (e.g. traffic incident, fire, shooting or bombing, etc.). For example, a traffic accident can take place at a location while people are rioting there. Therefore, messages should be grouped into different clusters based on the types of incidents too. The possibility of multiple incidents (of same or different category) happening at a location at the same time depends partly on the area (size) of the location and the span of time. For example, London is a big area; so, many incidents of the same type can concurrently take place there. However, it is less likely that more than one incident of the same type will happen concurrently at Oxford Street (in London), which is about 2 Km long. If the area of a location is even smaller e.g. the Tower Bridge, then the chance is almost zero. While clustering messages, the messages are grouped according to the following factors, in the given order, to get a picture of a situation:

1. Location of Incident
2. Category of Incident
3. Actor(s) involve in the incident

For example, if we receive ten messages and five of them contain information about a place A, three of them contain information about a place B and the rest are about place C, then we can analyse those five messages in order to find out what is happening in place A. Three out of these five messages, for example, may report a traffic incident while two other messages may say there was a fire (a car was on flame). However, by looking at the actors involved in the incidents, it may appear to us that all messages are, in fact, reporting (or not reporting) the same incident. Thus, it may be possible to get a picture of a situation at a given place by analysing the messages this way.

I have developed an application tool (written in C#, which uses XPATH to query the XML data) that clusters messages following the above procedure and produces



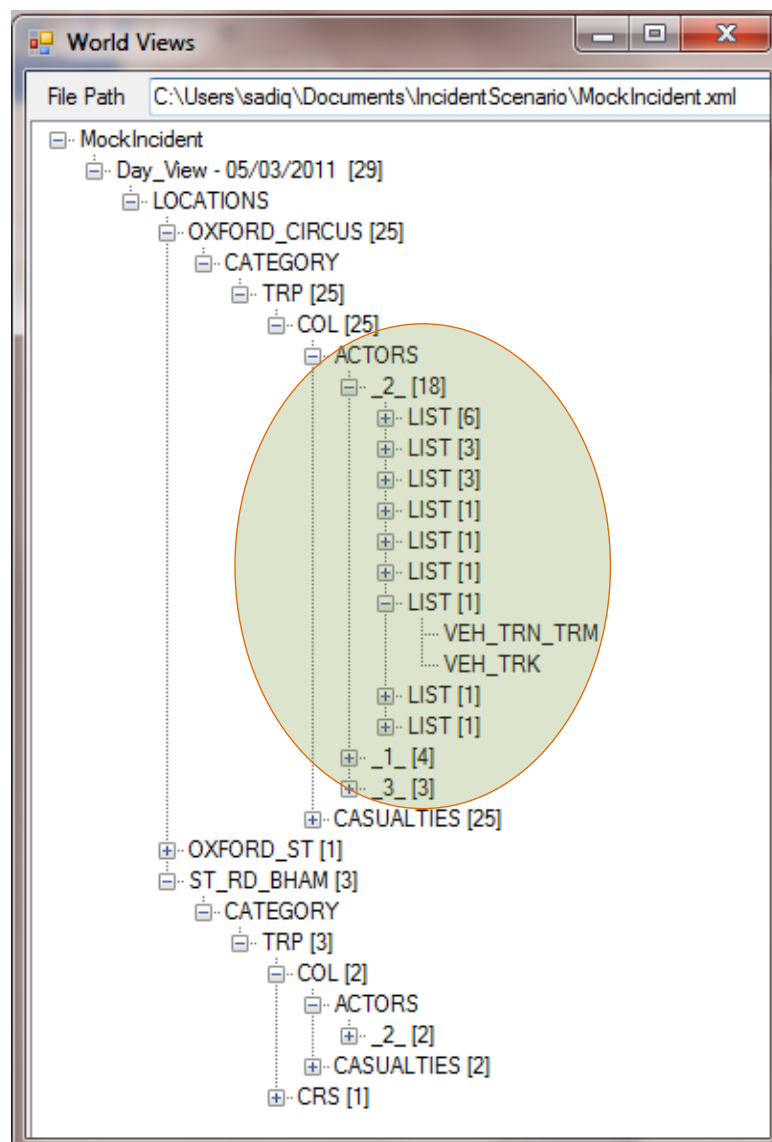


Figure 7.2: Worlds in Tree View: Messages are grouped according Location of Incident first

different views of one or more situations as shown in Figure 7.2. The application also allows its users to specify the order in which they want to cluster the messages. Categorising the messages in a different order may, sometimes, give a better view of a situation. For example, imagine we have received the following two messages:

1. Traffic collision near Oxford Circus, London
2. Major road accident at Oxford Street, London

If we categorise the messages in the consecutive order of Location, Category and Actor, these two messages may get listed under two different places (although Oxford Circus is on Oxford Street). However, if the messages are grouped according to the Category of Incident first and then according to the Location and Actor consecutively, as shown in Figure 7.3, these two messages will be shown in the same category (Transport incident) and it may be easier to relate them to the same incident even though they are listed under different locations.

After analysing the messages (TSO), the application indicates how many messages claim one or more incidents have taken place at a certain location and how many of them agreed upon a claim that a certain type of incident has taken place in that location. The number of messages is given in square brackets as shown in Figure 7.2. The application presents the Actors in descending order based on their frequency of occurrence. Figure 7.2 shows that eighteen (18) messages (TSOs) claim that there were two actors involved in the incident as opposed to four (4) and three (3) messages claimed the number of actors involved in the incident were one (1) and three (3) respectively. Actors that appear in the same message have been listed in the same ‘LIST’. The number in square brackets next to each ‘LIST’ shows how many messages mentioned the set of actors included in that ‘LIST’. Figure 7.2 shows, one (1) message claimed that the actors were **Tram** (VEH\_TRN\_TRM) and **Truck** (VEH\_TRK). Figure 7.4 shows, **Train** (VEH\_TRN) and **Truck** (VEH\_TRK) have been listed under **LIST [6]**, which means, six messages have stated that the pair of actors involved in the incident (transport collision) are **Train** (VEH\_TRN) and **Truck**. Since the list (**LIST [6]**) comprising of train and truck has been mentioned by maximum number of messages (6 times), **LIST [6]** has been shown on top of other lists (e.g. LIST [3], LIST [3], LIST [1], etc.). A user can also find out how many messages mention a particular actor by simply double clicking on that actor. Figure 7.4 shows on its right side that Truck has appeared as an actor in 15 messages (along with its counterparts).

The advantage of this approach is that the view of a situation that we get from this particular approach depends on how deep down the tree we are looking at. That is, while it produces clusters of totally consistent and relevant messages (world views) at the lowest levels of the tree, it also allows a user to view larger clusters of messages that are relevant but not totally consistent with each other, by looking at upper levels of the tree. For example, if we look at the top level of the tree, then we get information about multiple incidents that took place in different locations on the same day (and possibly same time). If we go one step down, then

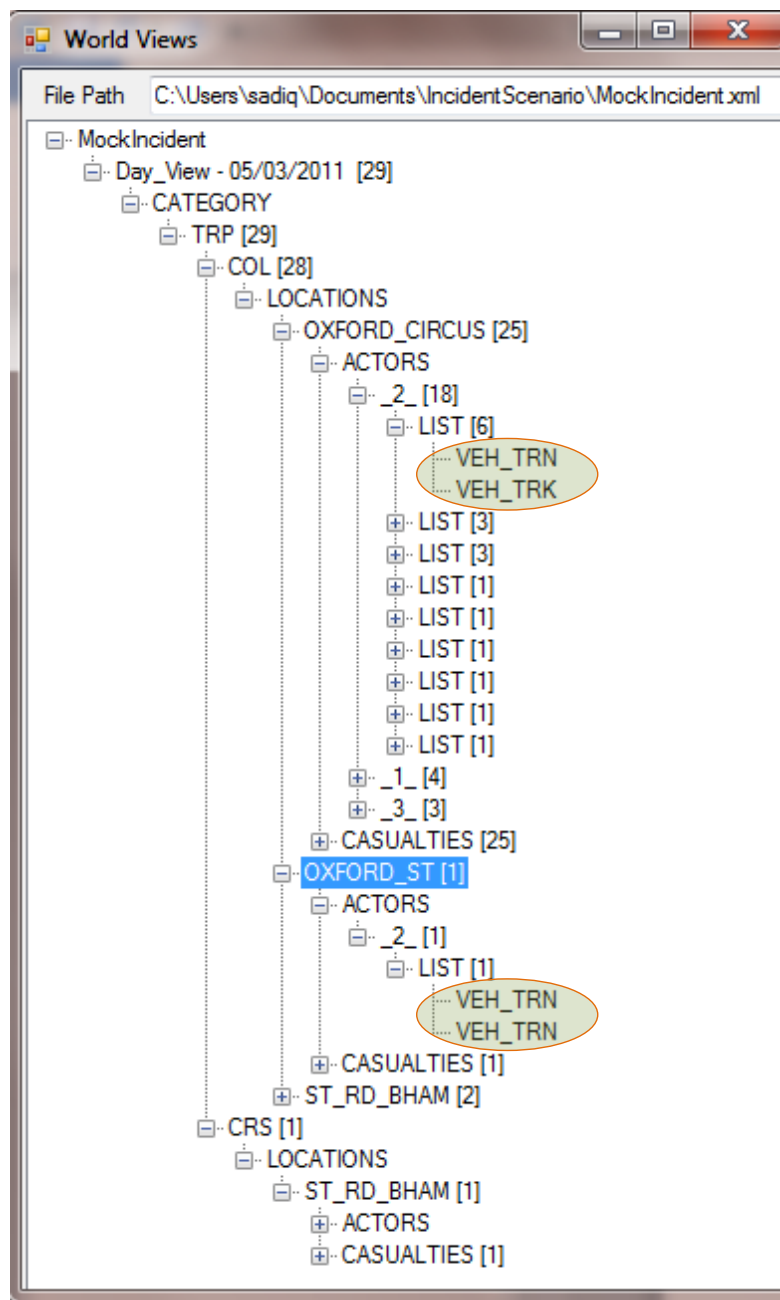


Figure 7.3: Worlds in Tree View: Messages are grouped according to Category of Incident first

we get the information of multiple incidents of different types (e.g. Traffic Incident) that took place at each of the locations. If we go one more step down, then we get a narrower view (e.g. traffic collision, break down, crash, etc.). If we go further down, then it says how many actors (e.g. vehicles) were involved in an incident (, as per the claims made in the messages). This dual-view feature of this tool may

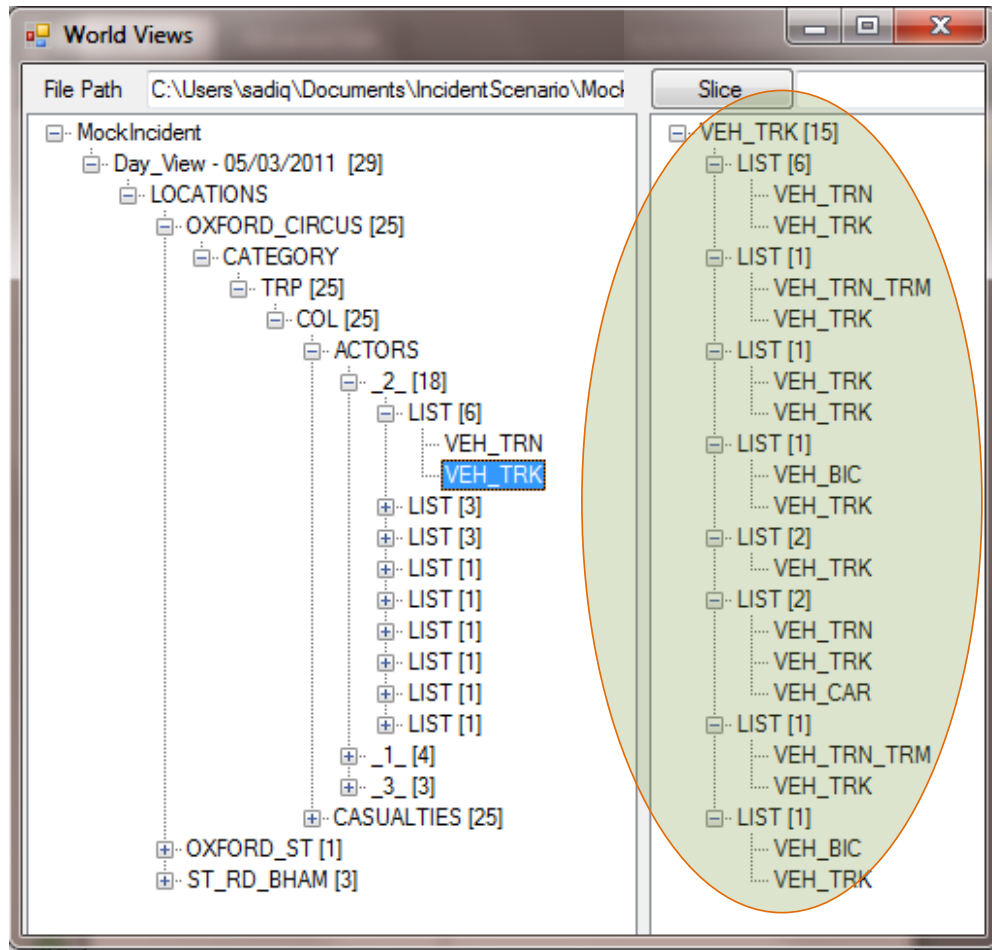


Figure 7.4: All Occurrences of a Selected Actor

be useful for a decision maker to make decisions in some cases. However, this kind of analysis has limitations and the objective of this research is to create consistent world views of a situation that are logically sound. This method of analysis cannot judge the logical soundness of a statement. For example, consider the following three messages:

- A car has collided with an aeroplane at Oxford Street.
- How can a car collide with a plane at Oxford Street? (i.e. No collision between car and plane at Oxford Street)
- Car cannot collide with a plane outside an airport.

The first message in the above list is an absurd and illogical statement although it may be argued that this type of messages may seem perfectly logical in some

scenarios, particularly if a message is sent as a hurried tweet. If we think that a message like "A car has collided with an aeroplane at Oxford Street." should be taken seriously, then we are likely to create false alarms as many a time people knowingly make this type of statements in jest or in sarcasm. If this type of sentences really have to be taken as sincere and seriously considering that the mistake was made for writing in a hurry, then we will have to correct the sentence to make it meaningful. Since it is possible to write many correct sentences by changing one or two words from each of such sentences, this will increase the complexity manifold. For example, which one of the following sentences shall we take as factual to replace the original (absurd) sentence "A car has collided with an aeroplane at Oxford street":

"A car has collided with an aeroplane at Oxford airport" or

"A car has collided with an antelope at Oxford Street" or

"A car has collided with an antelope on an Oxford Street" or

"A car has collided with another at Oxford Street"

This type analysis is beyond the scope of this thesis. Hence, I consider the messages depending on how they appear (on their face value), in order to keep the problem tractable. The analysis or clustering method described above cannot identify illogical statements like 'A car has collided with an aeroplane'. This method, at best, can say (as shown in Figure 7.5) that there are three messages about a car and a plane; one of them says there was a collision between them, while other two messages refute it. It is this limitation that I seek to address by adopting logic and constraint satisfaction.

## 7.2 Modelling and Consistency Checking with Alloy

Alloy is a model specification language based on first-order logic and set theory [92]. Alloy specifies objects (e.g. person, vehicle, etc.), properties of objects (e.g. *name* of a person, *make* and *model* of a vehicle) and their relations (relations between objects and relations between an object and its properties). An Alloy model also includes constraints and expressions (Facts, Predicates, Assertions, etc.) [64].

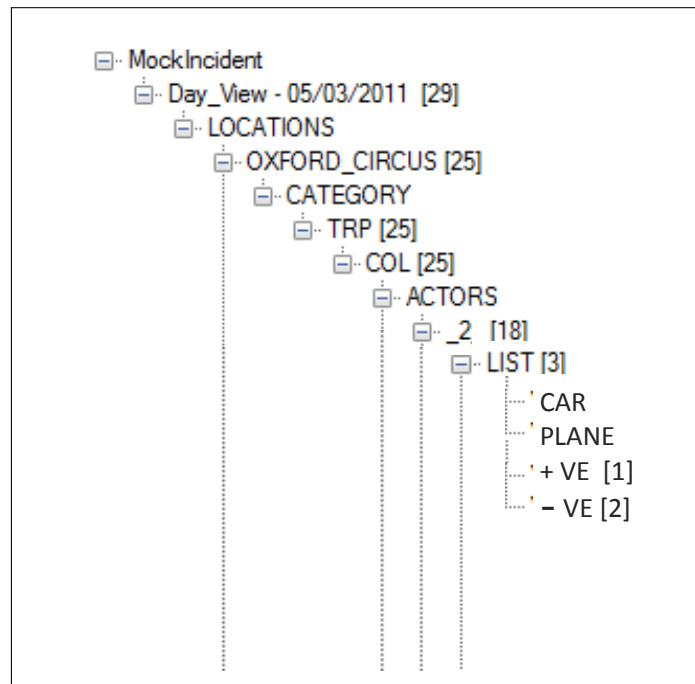


Figure 7.5: Contradictory Messages in Tree View

### 7.2.1 Alloy Language

Each Alloy model starts with the keyword **module** followed by its name, which is referred to as the *module header*. The building blocks of an Alloy model are Signatures (sets), Facts (constraints), Predicates and Functions (expressions). Each Signature (labelled with the keyword **sig**) represents a set of objects. A signature is similar to a ‘class’ definition in Object Oriented Programming. Below are examples of three *signatures*, namely ‘FileName’, ‘File’ and ‘Folder’.

```

sig FileName {}

sig File {
  name: one FileName,
  location: one Folder
}

sig Folder {
  file: set File,
  subFolder: set Folder
}

```

Two of the signatures, *File* and *Folder*, have two fields each. However, the other

signature, *FileName*, does not have any field. In this example, the ‘name’ and ‘location’ fields of the signature, *File*, specify that a file must have only one name and one location. Facts, functions and predicates are labelled with **fact**, **fun** and **pred** respectively. A fact specifies a constraint that is always assumed to hold. A predicate defines a reusable (i.e. parametrised) constraint. A function defines a reusable expression that returns a value. Figure 7.6 shows the anatomy of a basic Windows File System Model written in Alloy (that assumes that different types of file will be in different folders i.e. all files in each folder will be of same type).

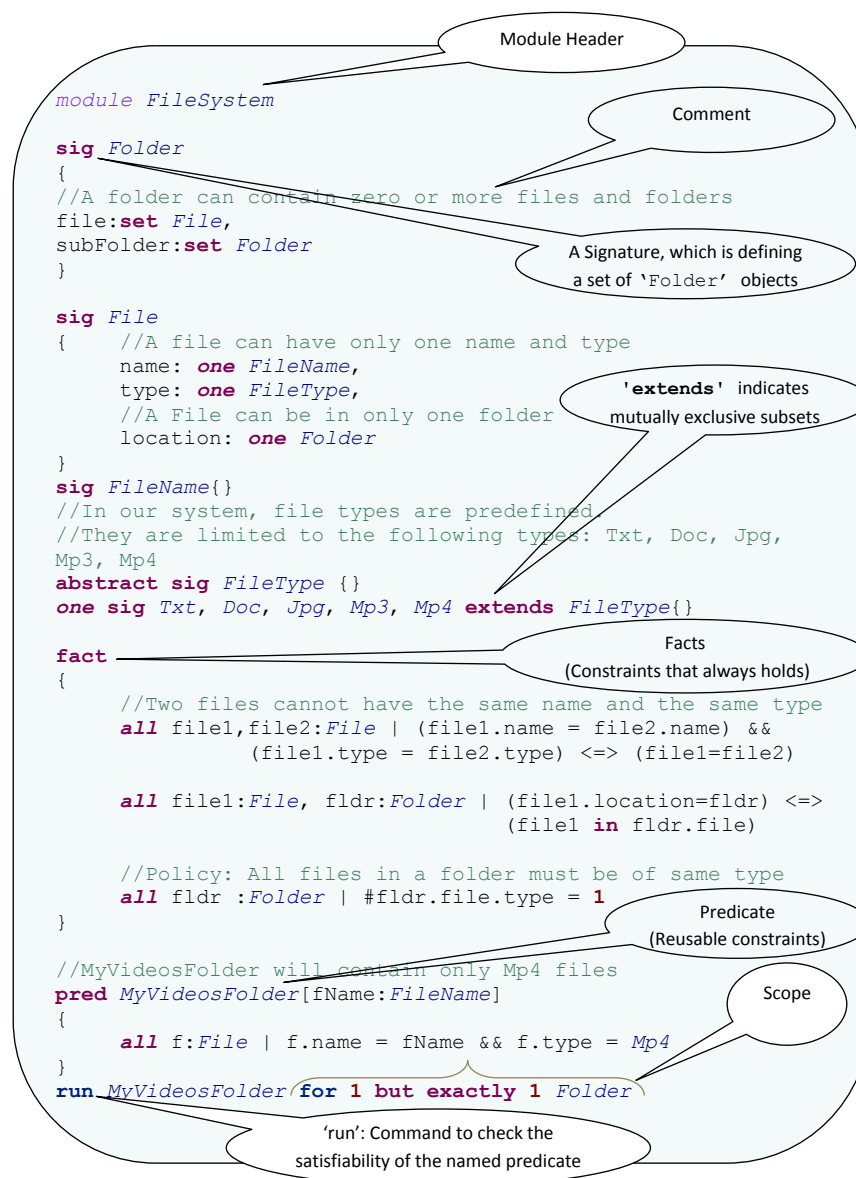


Figure 7.6: A File System Model in Alloy

### 7.2.2 Alloy Analyzer

Alloy's tool, the Alloy Analyzer [92, 64], is a model finder that finds a model by analysing specifications written in the Alloy specification language. Alloy Analyzer takes the constraints of a model and tries to find an instance that satisfies them within a given scope (see Section 7.2.3 for more details on *scope*). Alloy Analyzer internally uses a SAT solver to check if a predicate is satisfiable and returns true or false based on the predicate's satisfiability. However, a negative result given by the Alloy Analyzer is not absolute, because it examines only a finite space of cases based on the given scope, and therefore the analysis is not complete.

A basic Integrated Development Environment (IDE) is provided for manually writing new Alloy models, opening and editing existing models and executing different commands for finding satisfying instances of predicates and assertions. Satisfying instances of a model can also be graphically viewed (as shown in Figure 7.7) and customised in the Alloy Analyzer.

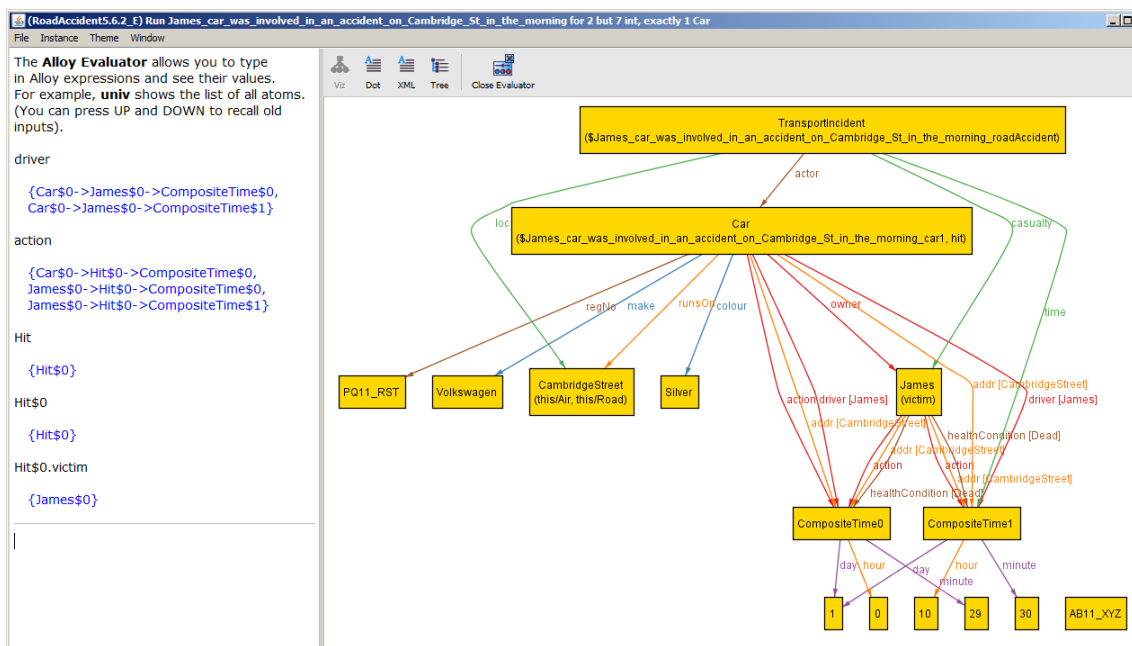


Figure 7.7: Instance of an Alloy Model

### 7.2.3 Scope

Since Alloy Analyzer tries to find an instance of a model within a given scope, a predicate that is unsatisfiable (i.e. no satisfying instance can be found) with



one scope may be satisfied in a larger scope. For example, consider a predicate expressing:

“The driver of a car lost control and hit another car”.

At least 1 person and 2 cars are needed to satisfy this predicate i.e. the **minimum scope** required to satisfy the above predicate is 1 person and 2 cars. Hence, Alloy Analyzer will find it unsatisfiable if it tries with a scope 1 person and 1 car. Alloy Analyzer may start with a minimum scope; however, it has to keep on searching for a satisfying assignment by gradually increasing the scope of each variable. The smallest scope that satisfies a predicate is referred to as a “minimal scope”. In many cases, the minimum scope required to satisfy a predicate is unique in a given context (i.e. with a given set of facts and assumptions). For example, based on the facts that ‘Blue is not Red’ and ‘a car cannot hit itself’, it is necessary to have at least two cars in the universe for the predicate ‘a blue car hit a red car’ to be satisfiable. However, there may be some predicates that may have more than one minimum scope. For example, there may be a predicate  $P$ , which has two variables  $A$  and  $B$ . Predicate  $P$  can be satisfied in a state with 1 $A$  and 2 $B$ s, and  $P$  can also be satisfied in a state with 2 $A$ s and 1 $B$ . The scopes of the variables in both of these states need to be recorded (stored in the database) as minimum satisfying scopes. A predicate along with each of its minimum satisfying scopes may be referred to as a variant of that predicate. For example, if a predicate  $P$  has two minimum satisfying scopes  $S_1$  and  $S_2$ , then the predicate  $P$  is said to have two variants  $P_{S_1}$  and  $P_{S_2}$ . The world view construction procedure (explained later in this chapter) will treat each these variants as a distinct predicate, except that none of the world views will contain more than one variant of an original predicate. Suppose a predicate  $P$  has two minimal scopes  $S_1$  and  $S_2$ . During the process of constructing world views, predicate  $P$  along with its scopes should be treated as two different predicates  $P_{S_1}$  and  $P_{S_2}$ . However,  $P_{S_1}$  and  $P_{S_2}$  cannot appear in the same world view.

### 7.3 Processing Individual Messages

A message needs to be specified in Alloy specification language before the Alloy Analyzer can check its satisfiability. Therefore, every time a new message is received, a model of that message is constructed in Alloy.

## 7.3.1 Encoding TSO in Alloy

```

sig Object
{
  owner: set Person,
  addr: Location one -> set CompositeTime} //An object can be at a location in
// one or more moments in time but cannot be at two different locations at the
// same time. Therefore, addr is a relation that is mapping a single Location
// to zero or more moments in Time.

sig PassiveObject extends Object {}

sig ActiveObject extends Object
{
  action: Action set -> set CompositeTime
}

sig NonLivingObject extends ActiveObject{}

sig LivingObject extends ActiveObject
{
  healthCondition: HealthCondition one -> set CompositeTime //Only one condition
at a specific moment.
}

abstract sig HealthCondition {time: Lone CompositeTime}
one sig Well_N_alive extends HealthCondition {}
one sig SlightlyInjured extends HealthCondition {}
one sig SeverelyInjured extends HealthCondition {}
one sig Dead extends HealthCondition {}

```

Figure 7.8: Specification of Different Types of Objects in Alloy

It has been stated earlier that every message reporting about an incident is encoded into a TSO. Some of the important pieces of information contained in a message or TSO include Category (of Incident), Actor (involved in an incident), Location (of incident), Time (of incident) and Casualty. However, some of these items (elements) are made optional (in the TSO schema) to comply with situations when some information is missing. The following example demonstrates how different parts of a message map into different TSO components and subsequently encoded into Alloy.

Suppose we have received a TSO containing the following message:

“James had an accident with his car in Cambridge this morning (10am)”

The following objects need to be modelled in Alloy:

James, a person (Actor → TSO/EVENT/ [[ETYPE]/ACTOR])	<pre>sig Person extends LivingObject {     uses: set Object,     home: set Home,     wound: set Injury } sig Man, Woman, Boy, Girl extends Person {}  one sig James extends Man {}    /*There's only one man whose name is James in our universe */</pre>
Cambridge, a location (Event-Location → TSO/EVENT/ [[EGEO]/POSITION])	<pre>sig Location {}  sig London, Oxford, Cambridge extends Location {}  sig Road, RailTrack, Air, Waters in Location{}</pre>
10am this morning, a date and a time (Time of Event → TSO/EVENT/ [OCC_DATIME])	<pre>sig CompositeTime {     day: Int,     hour: Int }{ day &gt; 0 &amp;&amp; day &lt;=31 &amp;&amp; hour &gt;=0 &amp;&amp; hour &lt;=23}</pre>
Accident (Event → TSO/EVENT)	<pre>sig Event {     loc: one Location,     time: some CompositeTime,     actor: some ActiveObject,     casualty: set LivingObject }{loc = actor.addr.time &amp;&amp; loc = casualty.addr.time} //actor's //location at the time of incident has to be the same as event //location.  sig TransportIncident extends Event{ {     some actor &amp; Vehicle    //It requires at least one Vehicle     loc in Road + RailTrack     one time }  sig Breakdown extends TransportIncident{#actor = 1} sig Collision extends TransportIncident{#actor = 2} sig Crash extends TransportIncident{#actor = 2}</pre>

Figure 7.9: Specification of Other Extended Objects in Alloy

- Accident, an event ( Corresponds to the TSO element EVENT, *TSO/EVENT*)
- James, a person (Corresponds to the TSO element ACTOR, *TSO/EVENT/[[ETYPE]/ACTOR]*)
- Car, a vehicle (Corresponds to the TSO element ACTOR, *TSO/EVENT/[[ETYPE]/ACTOR]*)
- Cambridge, a location (Corresponds to the TSO element POSITION

*TSO/EVENT/[[EGEO]/POSITION])*

- 10am this morning, a date and a time (Corresponds to OCC\_DATIME  
*TSO/EVENT/[OCC\_DATIME])*)

The TSO elements corresponding to the items listed above are given in brackets, along with their parent elements. It should be noted that the optional elements of TSO are enclosed in square brackets. For example, ETYPE and its child elements are optional elements of EVENT. However, ACTOR is a required element of ETYPE.

<p>Car, a vehicle (Actor → TSO/EVENT/ [[ETYPE]/ACTOR])</p>	<pre>sig Vehicle extends NonLivingObject {   make: one VehicleManufacturer,   regNo: one VehicleRegistration ,   colour: one Colour, //suppose no car has more than one colour in our universe    driver: Person Lone -&gt; set CompositeTime, /*There can be only one driver at a given time. Although the same driver can drive a vehicle at multiple moments in time but multiple driver cannot drive at the same moment in time. At times, there may be no driver in the vehicle at all. */   runsOn: some RailTrack+Road,   hit : Hit } { addr.CompositeTime in runsOn}  sig Train, Tram, Bike, Car, Truck, Lorry extends Vehicle {}  abstract sig Colour {} one sig Black, Blue, Green, Red, Silver extends Colour {} abstract sig VehicleManufacturer {} one sig BMW, Ford, Honda, Nissan, Toyota, Vauxhall, Volkswagen, Volvo extends VehicleManufacturer {}  abstract sig VehicleRegistration {} one sig AB11_XYZ, AB22_XYZ, AB33_XYZ, PQ11_RST, PQ22_RST, PQ33_RST extends VehicleRegistration {}</pre>
--	---

Figure 7.10: Specification of a Car and other Vehicles in Alloy

We need to model a generic object before we model Person and Car because both of them are different types of objects. One is a **living object** and the other is a **non-living object** while both of them are active objects. Those objects that can do things on their own like hitting another object are referred to as an **active object**. The object which is not active is called a **passive object**. The definition and interrelation of these objects specified in Alloy are shown in Figure 7.8. Specification of other elements referred to in the message is shown in Figure 7.9 and

Figure 7.10. Once we have the building blocks correctly specified, we can encode the TSO (the message itself) as a predicate (Figure 7.11). An Alloy predicate that contains only one message is referred to as a **simple predicate** and a predicate that combines multiple simple predicates (i.e. contains multiple messages) is referred to as a **compound predicate**.

It is important to note that most of the entities (e.g. man, car, etc.) referred to in a TSO are likely to be existentially quantified although there may be some constants that refer to definite entities like geographical locations (e.g. Oxford, Cambridge, etc.). The existentially quantified elements will be Skolemized (i.e. the existential quantifiers will be replaced) with new constants. Distinct Skolem constants (from the same or different TSOs) may, however, denote the same entity in some satisfying model.

```

pred James_had_an_accident_with_his_car_in_Cambridge_this_morning
  [roadAccident:TransportIncident, car1: Vehicle]
{
  roadAccident.actor = car1
  car1.owner = James
  roadAccident.time.day = 1
  roadAccident.time.hour = 10
  roadAccident.loc = Cambridge
}

```

Figure 7.11: An Alloy Predicate

### 7.3.2 Internal Consistency Checking

Since the messages need to be encoded into TSO before performing the consistency analysis and the TSO has a well-formed structure and a limited or finite number of vocabulary for encoding messages in the context of crisis and disaster management, the messages that cannot be encoded into TSO, are either ill-informed (not well-formed) or irrelevant (i.e. beyond the scope of the TSO) or both. Hence, these messages may be discarded automatically. Thus, the TSO format ensures that the messages are sufficiently well-formed and they abide by the rules (facts and/or physics) set out for a specific context.

When a TSO is modelled or encoded in Alloy, the model is augmented with two kinds of facts:

- a) World physics, which are always true (e.g. the same object cannot be at more than one location at the same time) and
- b) Information or assumptions that are considered to be true in a given context (e.g. 22:00 GMT means that it is night-time and dark in the UK).

Some previously known and reliable information may also be used as a fact. For example, an unambiguous piece of information reliably collected from a sensor or a CCTV camera, or the information about drivers and vehicles held in the DVLA (Driver and Vehicle Licensing Agency in Great Britain) database may be treated as facts.

Once the facts are known to the system, the Alloy Analyzer is used to check the internal consistency of each message i.e. to check whether the message is consistent with the known facts and world physics<sup>2</sup>.

ID	Predicate Name	Satisfiable	Scope
A	Predicate-1	True	1 Person, 2 Cars
B	Predicate-2	True	2 Persons, 2 Cars
C	Predicate-3	False	2 Persons, 2 Cars
D	Predicate-4	True	2 Persons, 1 Car, 1 Bike
E	Predicate-5	False	10 Persons, 1 Train
F	Predicate-6	True	2 Person, 1 Car
G	Predicate-7	True	... ..

Figure 7.12: Individual Messages Stored in Database with their Truth Values

The satisfiability of the message i.e. the boolean result (True or False) returned by the Alloy Analyzer is stored in the database, as shown in Figure 7.12, along with a record of the satisfying scope. If the Alloy Analyzer cannot satisfy a predicate (message) within the minimum scope, it will carry on searching for a solution by

<sup>2</sup>Although, in principle, the appropriate set of facts (constraints) should be reloaded before processing each new message, I have not done so. However, this is one of the topics for my future research.

gradually increasing the scope of each variable or object. The search stops when the scope allows a distinct assignment to every constant in the Skolemized formula. Some messages may be internally consistent; while there may be some messages that are intrinsically inconsistent i.e. they do not satisfy individually. For example, messages like “A car is travelling at 700 km/h” will never satisfy the constraints if we know that ‘even the latest cars cannot run at a speed of 700km/h’.

## 7.4 Construction of Consistent World Views

It has been stated earlier that after checking the satisfiability of each individual message and the result (i.e. whether they are consistent with the known facts) is stored in the database, as shown in Figure 7.12, a world view is constructed by combining the available messages together if their conjunction is found satisfiable too. Maximal consistent groups of messages form world views as each group produces a picture of a possibly prevailing situation of our ‘world’. For example, if there are seven messages A, B, C, D, E, F and G and they are all satisfiable together, then

**A B C D E F G**

would form the world view. However, as it has been noted above, there may be some messages that are inconsistent with our axioms. Therefore, it will not be possible to construct a world view by combining other messages with these unsatisfiable messages i.e. no compound predicate (cluster of messages) that includes an intrinsically inconsistent message will satisfy. Hence, filtering out the unsatisfiable messages will save a significant amount of time in assessing world views. Thus, the biggest possible world view constructed with the messages shown in Figure 7.12 could be **A B D F G** as the messages C and E are intrinsically unsatisfiable. However, if the predicate A has two variants  $A_{S1}$  and  $A_{S2}$  for having two minimum satisfying scopes  $S1$  and  $S2$ , as explained in Section 7.2.3, then the biggest possible world view would be either **A<sub>S1</sub> B D F G** or **A<sub>S2</sub> B D F G**.

### 7.4.1 Processing with a minimum scope

The minimum scope required to satisfy a simple predicate has been discussed in Section 7.2.3. Similar to the case of a *simple predicate*, I start checking the consistency of a *compound predicate* (defined in Section 7.3.1) with its minimum scope, although compound predicates may not always satisfy within their minimum scope. As I am going to define the minimum scope of a compound predicate next, re-

call the definition of the minimum satisfying scope of a simple predicate and note that the minimum satisfying scope for each simple predicate was recorded in the database. The *Minimum Scope*, which is likely to satisfy a compound predicate, is the pointwise maximum of the minimum scope of individual messages contained in the predicate. Consider the following two messages as an example:

1. “A stationary car was hit by another car” [Minimum Scope: 1 Person, 2 Cars]
2. “There was a traffic accident” [Minimum Scope: 1 Person, 1 Car]

The minimum scope required to satisfy the first message (predicate) is 1 Person and 2 Cars. The minimum scope required to satisfy the second message is 1 Person and 1 Car. So, the pointwise maximum scope for ‘Person’ is 1 and the pointwise maximum scope for ‘Car’ is 2. Thus, the minimum scope required to satisfy the compound predicate consisting of the above messages is 1 person and 2 cars.

However, a compound predicate may not always satisfy with the *Minimum Scope*; it may require a larger scope to satisfy. Consider the following messages:

- “A blue car hit a red car at Trafalgar Square” [Minimum Scope: 1 Person, 2 Cars]
- “A green car hit a red car Trafalgar Square” [Minimum Scope: 1 Person, 2 Cars]

They will not satisfy with their *Minimum Scope* of Car (which is 2). Hence, the scope of car needs to be increased. These two messages will satisfy with the availability of 3 Cars and may refer to an incident, a pileup involving three cars, similar to what is shown in the picture (Figure 7.13). However, if we

need to increase the scope for satisfying a compound predicate, a good bound on the *Maximum Scope* that we may need to consider is the pointwise sum of the minimum scope of individual messages. Hence, for



Figure 7.13: A Pileup

the above two statements, the pointwise sum of the minimum scope for ‘Person’ and ‘Car’ are  $(1 + 1 =) 2$  and  $(2 + 2 =) 4$ , respectively. However, with maximum scope all the messages in a cluster (compound predicate) will satisfy together unless they make contradictory statements about some definite identifier. The above two statements are consistent if we allow four cars and two distinct incidents, but they are perhaps more likely inconsistent reports of a single incident with two cars.



**World View Construction Process:**

If **ABDFG** is a compound predicate and it does not satisfy with the minimum scope, then we try all of the following combinations of the messages it contains:



Each of the satisfying combinations forms a world view and the unsatisfying combinations will be split further to check if they form any smaller world views.

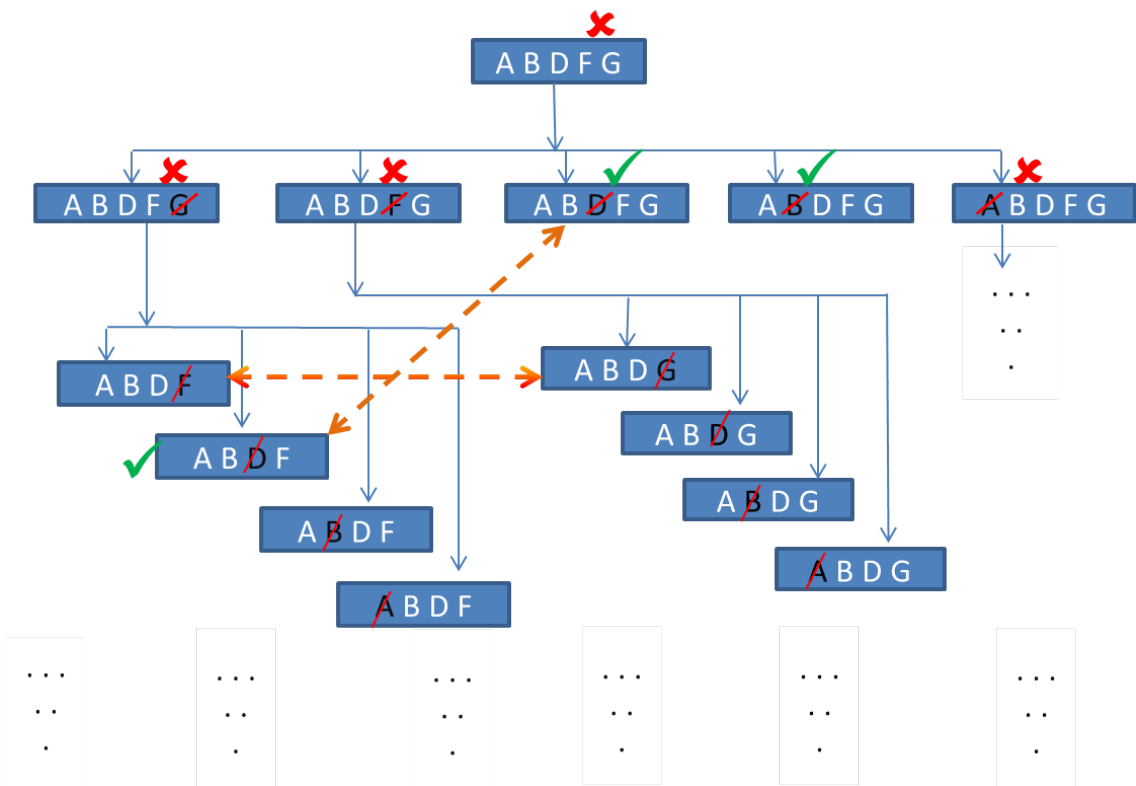


Figure 7.14: World View Construction Process

This process (illustrated in Figure 7.14) continues until it comes to the point that none of the messages are consistent with each other and we get world views with a single message in each of them. This ensures that we get the complete list of possible world views. This operation of generating message-combinations and checking their satisfiability by feeding each combination into the Alloy Analyzer is performed by the Consistency Analysis and Conflict Resolution unit (tagged 5 in the system architecture, shown in Figure 5.1 and Figure 7.1). Hence, the Consistency Analysis and

Conflict Resolution unit has mainly two internal components: Message-Cluster Generator and Model Finder (Alloy Analyzer). The overall process of generating world views is shown in Figure 7.15. The process shown in Figure 7.15 involves a breadth-first search for finding world views. The process of finding a world view which involves depth-first search is shown in the appendix in Figure B.1 (Appendix B).

During the whole process, care needs to be taken to avoid checking the satisfiability of the same element twice. The first occurrence of a cluster is checked for satisfiability but all subsequent occurrences are ignored. Also, if a larger world view is found, then there is no need to consider any of its subsets as a world view.

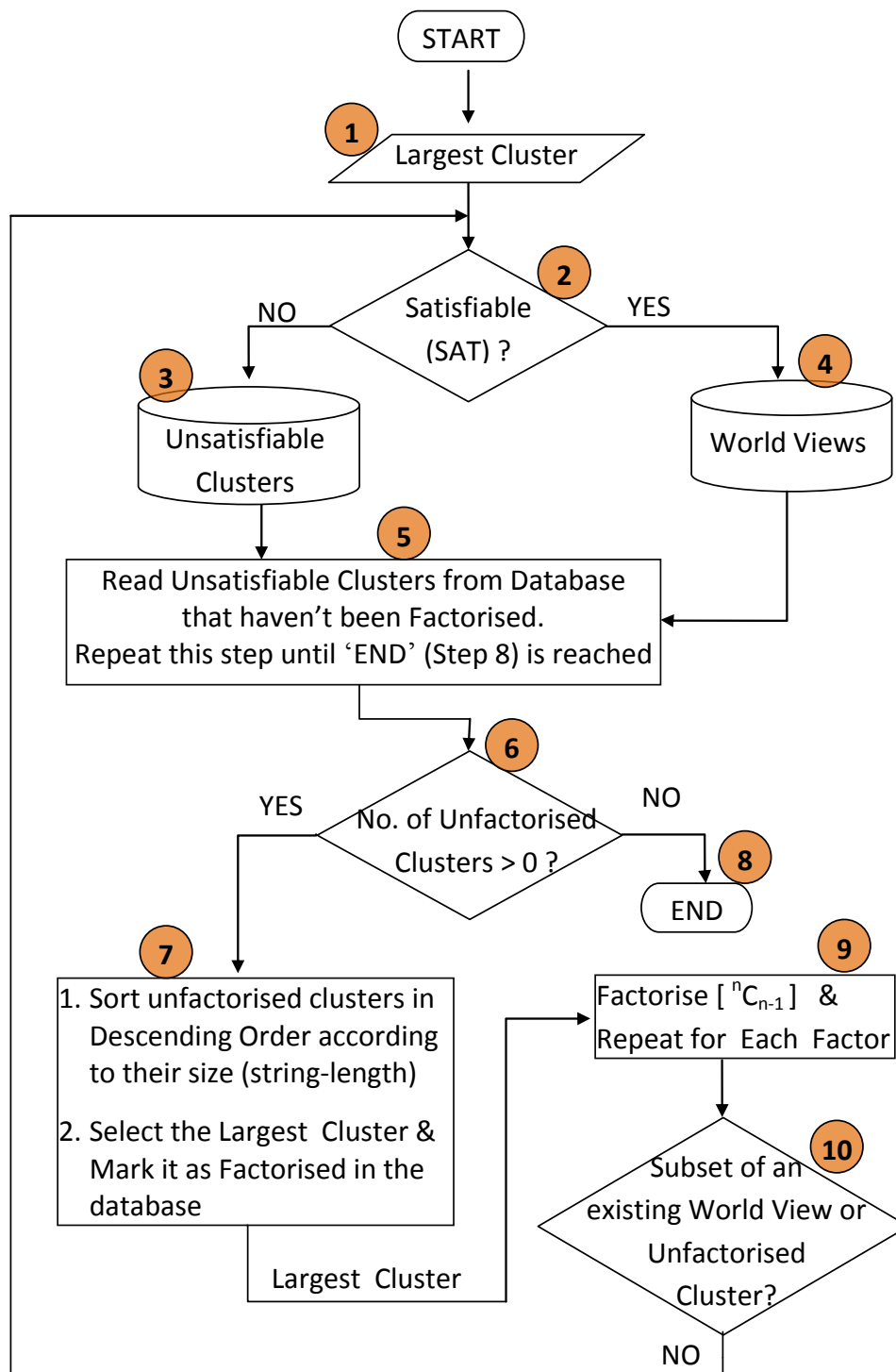


Figure 7.15: Process Diagram of the World View Generator (Breadth-First Search)

For example, if we get the world view ABFG, then we will disregard all of its subsets e.g. ABF, ABG, etc. If, in any case, a larger world view is found that is a superset of one or more existing smaller world views, then the only larger world view is kept by discarding all its existing subsets (smaller world views). Hence, in a worst case scenario when none of the messages in a cluster of  $n$  number of messages are consistent with each other, a total of  $\sum_{i=1}^n {}^n c_1$  distinct combination of messages will have to be checked for satisfiability.

### Illustration of the Process Diagram in Figure 7.15 with an Example

Suppose we have four messages A, B, C and D. Hence, the largest possible message-cluster is ABCD. Suppose ABD and AC are the two world views that can be constructed from these four messages.

Step - 1: ABCD

Step - 2: No

Step - 3: ABCD is stored in the database as unsatisfiable cluster (, which has not been factorised yet)

#### **Iteration - 1:**

Step-5: ABCD

Step-6: Yes

Step-7: ABCD

Step-9: ABCD is factorised into the factors ABC, ABD, ACD, and BCD.

Perform the following steps (Steps 10, 2 and 3 or 4) for each of these factors.

	<b><i>Iteration-1</i></b> <i>(ABC)</i>	<b><i>Iteration-2</i></b> <i>(ABD)</i>	<b><i>Iteration-3</i></b> <i>(ACD)</i>	<b><i>Iteration-4</i></b> <i>(BCD)</i>
Step-10	No	No	No	No
Step-2	No	Yes	No	No
Step-3/4	Step-3 (Saved as an Unsatisfiable Cluster)	Step-4 (Saved as a World View)	Step-3	Step-3

**Iteration - 2:**

Step-5: ABC, ACD, and BCD

Step-6: Yes

Step-7: ABC (is selected from ABC, ACD, and BCD)

Step-9: ABC is factorised into the factors AB, AC, and BC.

Perform Step-10 for each of these factors.

<b>Iteration-1</b> (AB)	<b>Iteration-2</b> (AC)	<b>Iteration-3</b> (BC)
<b>Step-10: Yes</b> (Subset of the World View ABD. No further action is taken)	<b>Step-10: Yes</b> (Subset of Unfactorised Cluster ACD. No further action is taken)	<b>Step-10: Yes</b> (Subset of Unfactorised Cluster BCD. No further action is taken)

**Iteration - 3:**

Step-5: ACD, BCD

Step-6: Yes

Step-7: ACD is selected from ACD and BCD

Step-9: ACD is factorised into the factors AC, AD, and CD.

Perform the following step(s) for each of these factors.

<b>Iteration-1</b> (AC)	<b>Iteration-2</b> (AD)	<b>Iteration-3</b> (CD)
<b>Step-10: No</b>	<b>Step-10: Yes</b> (Subset of Unfactorised Cluster ABD. No further action is taken)	<b>Step-10: Yes</b> (Subset of Unfactorised Cluster BCD. No further action is taken)
<b>Step-2: Yes</b>		
<b>Step-4</b> (saved as a World View)		

Iteration continues in the same manner, until the number of unfactorised clusters is found zero i.e. Step 6 returns 'No' and execution stops (Step 8).

### 7.4.2 Reprocessing with a larger scope and reliability of world views

If a cluster of messages does not satisfy within a given scope, it may satisfy in a larger scope. Therefore, if none of the messages are satisfiable together with a given scope or the user of this tool wants to see if the world view could be any different, then the user may re-run Alloy Analyzer on the same set of predicates by increasing the scope of the variables. However, these newly created world views may get less priority as they are likely to produce more ambiguous pictures of a situation. Consider the following three messages as an example:

1. “There is a collision between two cars” [Minimum Scope: 1 Person, 2 Cars]
2. “Head-on collision between two cars” [Minimum Scope: 2 Persons, 2 Cars]
3. “Man died as a car crashed on motorbike” [Minimum Scope: 1 Person, 1 Car, 1 Motorbike] (We are not certain that this car is not one of the cars stated in earlier messages)

The first two messages satisfy together and form a world view with only two men and two cars in the universe. If we increase the scope, all three messages will satisfy together and will form a bigger world view with only two men and two vehicles instead of two cars in the universe (as we are not certain that the car mentioned in the last message is not one of the cars stated in earlier messages). But this bigger world view gives a less conclusive message and is more ambiguous as there can be multiple possibilities of collisions between different vehicles. Therefore, this world view may be less dependable/credible. However, if the user is somehow convinced, for example, that there is more than one incident at a certain place then s/he may increase the scope without diminishing the credibility of the new world view. Also, in the case that the car mentioned in the third message was somehow uniquely identified and is one of the cars mentioned in message one and message two, the third message may be taken into the world view along with the first two without diminishing the credibility of the new world view. Also, smaller scope may not always produce a good result as two different incidents may appear as one incident with a smaller scope.

### 7.4.3 Impact of a new message on existing world views

Before explaining the impact of a new message on existing world views, it is important to describe the procedure of generating world views by incorporating the new

message. This will make it easier to understand the impact.

### Procedure of incorporating a new message

Instead of trying to create world views by combining the newly received message with all existing messages, the new message will be combined with each of the existing world views only. This will save the processing time by avoiding the repetition of the satisfiability checking of the same clusters that have already been checked before the existing world views were generated. The rest of the procedure will remain the same as that of creating the initial world views i.e. split them only when the new clusters do not satisfy. This whole process can be illustrated with an example shown

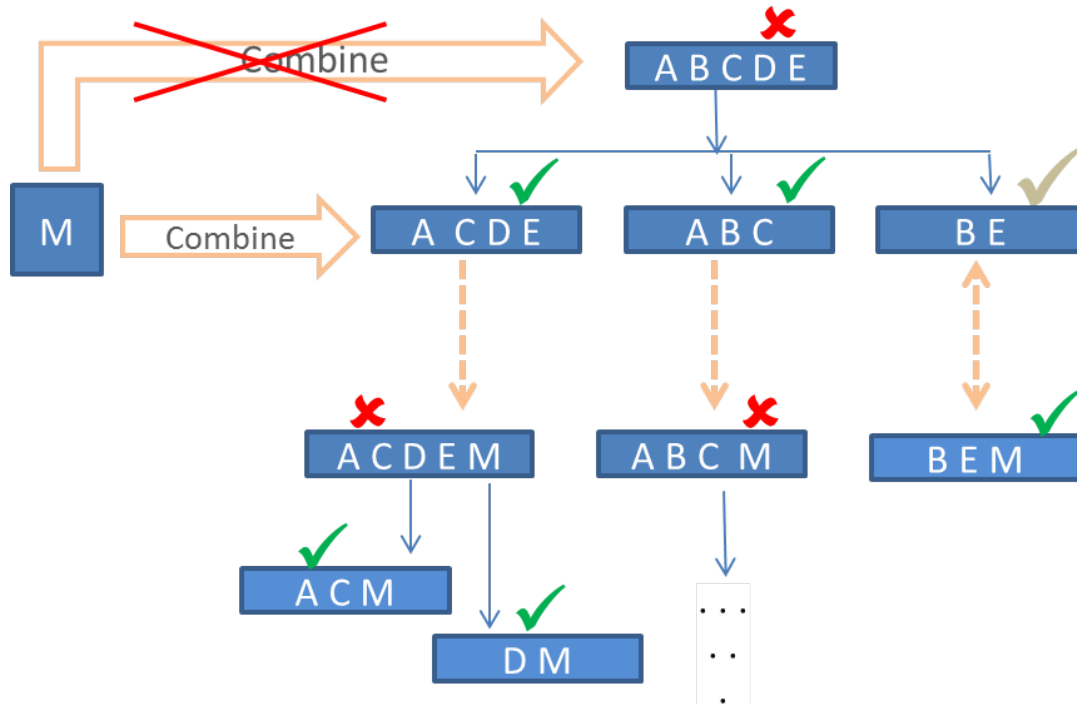


Figure 7.16: Combine a New Message with Existing World Views

in Figure 7.16. Suppose we have five messages **A**, **B**, **C**, **D**, and **E** and three world views namely **ACDE**, **ABC** and **BE**. Later, we have received a new message **M**. Instead of initiating the process of creating world views with the message cluster **ABCDEM** and downsizing it as necessary, we will start with clusters **ACDEM**, **ABCM** and **BEM**. We will repeat the same procedure (described earlier) of splitting them into smaller clusters if they are not consistent. Suppose at the end of the

process we have found three more world views namely **ACM**, **DM** and **BEM** in addition to the existing world views. Since the newly found world view **BEM** is a superset of the already existing world view **BE** and only the largest, unique and consistent sets of messages form world views, I will not consider **BE** as a world view any more.

### **Result of incorporating a new message**

When a new message is incorporated into the existing set of messages and world views, one or more of the following cases will arise:

1. None of the existing world views are consistent with the new message.
2. All of the existing world views are consistent with the new message.
3. Some of the existing world views are consistent with the new message and some are not. Those world views that are not consistent with the new message will lead to either of the following cases:
  - (a) None of their factors/subsets are consistent with the new message.
  - (b) Some of their factors/subsets are consistent with the new message.

Depending on which of the above cases hold when a new message is incorporated into the existing set of messages and world views, either of the following things will happen to the existing world views:

- In Case-1, all of the existing world views will remain unchanged and the new message will be treated as a separate world view on its own. Hence the number of world views will remain the same except the new ‘singleton’ world view.
- In Case-2, a new larger world view will be produced from each of the existing world views but the number of world views will remain the same.
- In Case-3, the number of world views will certainly increase because a new larger world view will be produced from each of the (existing) world views that are consistent with the new message, while the rest of the world views that are inconsistent with the new message will produce either of the two results:
  - a) No new world views will be generated as none of the factors/subsets of the (existing) world view are consistent with the new message. However, the original world views will be retained.



- b) One or more new but smaller world views will be generated (from one of the factors/subsets of the original world view are consistent with the new message) while the original world views will be retained.

## 7.5 World View Generation: Some Observations

Close observation of the world view generation process (described in Section 7.4) exposes some interesting phenomena of the world views that are outlined below.

The addition of a new message to the existing world views may or may not increase the number of world views. However, a new (additional) message cannot cause the number of existing world views to decrease, in any case. This is because if the addition of the new message does not produce any new world view, all the original world views remain unchanged (i.e. they are still world views). If the new message causes any change to the existing world views, then the number of world views either increases or remains the same (see Section 7.4.3 for further clarification). Figure 7.16 illustrates this phenomenon with an example. Each of the newer world views is either a subset or a superset of the old (original) world view which was used to produce them (i.e. that original world view itself or one or more of its factors are consistent with the message). However, generally, none of the world views are a subset or a superset of another world view. For example, Figure 7.16 shows, the addition of the new message,  $M$ , with the existing world view  $ACDE$  leads to two newer but smaller world views,  $ACM$  and  $DM$ , while the addition of  $M$  with the existing world view,  $BE$ , leads to a newer and bigger world view. Suppose  $M$  is the set of all messages that are used to create the world views and  $W$  is the set of all world views. Hence,  $W$  is a set of subsets of  $M$ . Since, none of the world views are a subset or a superset of other world views, the set of world views,  $W$ , is an anti-chain.

### Predictability of World Views

In some cases, the impact of a new message on the existing world views can be determined without even checking the satisfiability of the new clusters of messages. For example, if the new message ( $M$ ) is a logical implication (entailment) of another message (say  $A \rightarrow M$ ) and  $ABC$  is an existing world view (i.e.  $(A \wedge B \wedge C)$  is consistent), then  $ABCM$  will certainly be a world view (i.e.  $(A \wedge B \wedge C \wedge M)$  is also consistent). Likewise, if  $A \rightarrow \neg M$  then  $M$  cannot be included in any of the world

views which includes A.

If M is a tautology (or a contradiction) (i.e.  $\top$  or  $\perp$ ), then M can be included in all (or none) of the world views. However, including a message, which is a tautology, in the world views will not make the world views any more interesting.

### Maximum and Minimum Number of World Views

If all messages are consistent with each other, then we will get only one world view and that single world view will be of the largest possible size (as it will contain all of the messages). On the contrary, if none of the messages are consistent with each other, then we get as many world views as the number of messages i.e. each message will be treated as a world view. However, if some messages are consistent while some others are inconsistent with each other, then the number of world views will depend on how consistent or inconsistent the messages are with each other. Since the maximum number of clusters, none of which is a subset of another, that can be formed by choosing any number of items from a given set of  $n$  messages is  ${}^n C_{\frac{n}{2}}$ , the number of world views that may be generated from a set of  $n$  messages will range from 1 to  ${}^n C_{\frac{n}{2}}$  i.e.  $\frac{n!}{(\frac{n!}{2})^2}$ .

It should also be noted that the number of world views does not depend on the number of available messages; it depends on the mutual consistency of the messages instead.

### Distinct but Similar World Views

Since, according to the definition (given in the beginning of Chapter 5), world views need to be fully consistent, it is possible that we get large number of world views that are almost identical. The following messages give a good example of such cases.

1. "Collision between cars at Oxford Circus and one person died"
2. "Collision between two cars at Oxford Circus and two people died"
3. "Car accident at Oxford Circus. Drivers seriously injured."
4. "Collision between car and a motorbike at Oxford Circus and one person died"
5. "Collision between a car and a motorbike at Oxford and one person died"

All of the above messages report about (possibly the same) traffic incident. However, each of the five messages will create very similar but five different world views, as they differ in small things like number/nature of casualties and similar location-name. In some cases, we may get a better picture of a situation by considering the similar world views collectively, rather than separately. This leads to the discussion of consistent versus fuzzy world views, which I have discussed in Chapter 9 (Sect. 10.1).

## 7.6 Summary

The overall methodology of constructing world views from different messages has been described in this chapter. How different messages can be clustered to generate world views has been described in great detail. Some of the interesting phenomena of world views and their generation process including some predictability of world views and the minimum and maximum number of possible world views have also been outlined in this chapter. It should be noted that the messages are denoted or identified with letters (A, B, C....) from the English alphabet for clarity and simplicity. In a real life case, where a very large number of messages are likely to be processed, the message IDs may be defined with a code instead of a letter. For example, a message ID can be defined with Hexadecimal codes. If each message ID is represented with an 8-digit Hexadecimal code (e.g. FA9FBE0A), then a message cluster can contain billions of messages (up to FFFFFFFF or 4,294,967,295) in it. Thus, ‘F90AB013 FA9FBE05 AF5FDF90’ is an example of a message cluster containing three messages.

The well-formed structure of TSO makes it possible to perform logical and non-logical analysis of the messages in order to construct world views. While the non-logical analysis has limitations as highlighted in this chapter, the logical analysis also needs to be performed on the right set of data; otherwise it will not produce any good result. For example,

- “A tourist coach has crashed in Oxford”
- “Signal failure in London underground”
- “There are lot of people in Victoria Park”

these three messages are not contradictory but totally irrelevant. Hence, there is no point in checking the consistency of such irrelevant messages. Therefore, irrelevant

messages need to be separated before the logical consistency checking is performed. The success of the logical analysis also depends on determining the facts and physics of the world that define the constraints.

Alloy specification language and Alloy Analyzer, the tools for logical analysis of the messages, have also been introduced in this chapter. This knowledge of Alloy, Alloy Analyzer and the impact of the *scope* of each variable will help implementing the decision support system which will be described in the next chapter.

## CHAPTER 8

# Realising the Decision Support System

Implementation of different components of the decision support system and the process of encoding messages into TSOs (Tactical Situation Objects) have been described in this chapter. While the implementation of the data harvesting unit (Unit 1) shown on the system architecture diagram (Figure 5.1) has been described in Chapter 3, the implementation of the rest of the components of the system has been described successively in the following sections.

- Implementation of the TSO Encoder (Unit 2) and the Scoring Function (Unit 3) have been discussed in Section 8.1 and Section 8.2.
- Implementation of the Consistency Analysis and Conflict Resolution unit (Unit 5) has been described in Section 8.3 and Section 8.4. It has been stated in the previous chapter (in Section 7.3) that the logic based consistency analysis of the messages (TSO) is performed using Alloy and Alloy Analyzer.
- Some of the components of the system including the combiner function (Unit 4, 6 and 7) have been implemented in the back-end with SQL Server (using tables, queries, user-defined functions and stored procedures). However, a detailed description of their implementation have been omitted for the sake of brevity.
- The implementation of Unit 8, which generates and presents the ordered list of world views to the end-user and allows the user to apply the decision making policy, has been implemented in Section 8.5 and Section 8.6.

In addition to the detailed description of the implementation of different components given in respective sections (as outlined above), Table 8.1 gives a summary

on different components of the Decision Support System e.g. which of the components require manual processing and which of them are automatic, which part of the thesis discusses its theory and implementation, etc. The table also outlines the programming languages and third-party APIs used to develop different components of the DSS and the input and output (what the input/output data is, their data types and/or data source) of each component of the DSS.

Table 8.1: Summary on Different Components of the DSS

<b>Component ID</b>	<b>Name of the Component / Process</b>	<b>Where in Thesis it is Discussed</b>	<b>Automatic/Manual, I/O Data Format, and Language &amp; API used to Implement</b>
1.	Policy Driven Information Source Filter (Data Harvesting Unit)	Section 3.1 (Harvesting Open-source Data from Twitter)	<b>Automatic</b> <b>Input:</b> XML, JSON <b>Output:</b> Saves Input data into an SQL Server Database <b>Language used:</b> C# (ASP.Net)
2.	Tactical Situation Object Encoder	Section 6.5, 8.1	<b>Semi-automatic</b> <b>Input:</b> Information from the previous step (i.e. Data Harvesting Unit) stored in database. <b>Output:</b> TSO (XML) <b>Language used:</b> C# (ASP.Net)
3.	Policy Driven Scoring Function based on Provenance Factors	Section 8.2	<b>Automatic</b> <b>Input:</b> TSO (XML) and Database tables <b>Output:</b> Each message (TSO) is attributed with some scores corresponding to some trust factors and the scores are saved in an SQL Server Database <b>Language &amp; API used:</b> C# (ASP.Net), JavaScript, Google Maps and Geocoding API
4.	Scored Messages		

*Continued on next page*

Table 8.1 – Continued from previous page

<b>Component ID</b>	<b>Name of the Component / Process</b>	<b>Where in Thesis it is Discussed</b>	<b>Automatic/Manual, I/O Data Format, and Language &amp; API used to Implement</b>
	Conversion of Messages into corresponding Alloy specifications	Section 7.3, 8.3	<b>Manual</b> <b>Input:</b> Plain-Text / TSO (XML) <b>Output:</b> Alloy Specification saved in a (text) file with extension ‘.als’
5.	Consistency Analysis & Conflict Resolution	Section 7.3.2, 8.4	<b>Automatic</b> <b>Input:</b> a message or a message cluster written in Alloy Specification language <b>Language &amp; API used:</b> C#, Java and Alloy Analyzer
6.	Multiple Views of a Possible Situation	Section 7.4	(This is the output of the previous step.) <b>Output:</b> {Message/Message Cluster, Boolean} i.e. Each Message or Message Cluster is stored in Database along with its Satisfiability
7.	Combiner Function (Aggregate Score Calculator)	Appendix E	<b>Automatic</b> (Performed internally within MS SQL Server.) <b>Input:</b> Each message (TSO) and its associated scores corresponding to different trust factors that were saved in the SQL Server Database <b>Output:</b> Database tables are updated with aggregate scores. <b>Language used:</b> SQL (Stored Procedure)
8.	Decision Making Policy	Section 8.6, 5.1.6	Contrived policies are stored in database tables

*Continued on next page*

Table 8.1 – Continued from previous page

Component ID	Name of the Component / Process	Where in Thesis it is Discussed	Automatic/Manual, I/O Data Format, and Language & API used to Implement
9.	Weighted/Ranked Views of a Possible Situation	Section 8.5	<b>Output:</b> Google Chart (each world view is shown as a Bubble) <b>Language &amp; API used:</b> C#, JavaScript, AJAX and Google Chart

## 8.1 Encoding Messages into TSO

Since each TSO is essentially an XML file, constructing a TSO is, in fact, creating an XML file with situational information. Therefore, I have developed an application tool (in C#) that automatically constructs a complete schema for the TSO. The construction mechanism of the application tool and the TSO schema has been described in detail in Chapter 6 (Section 6.3). The complete schema constructed using my application tool enables the users to encode messages into TSO easily (although manually) using a standard XML editor. The construction of TSO using the auto-generated schema is described below.

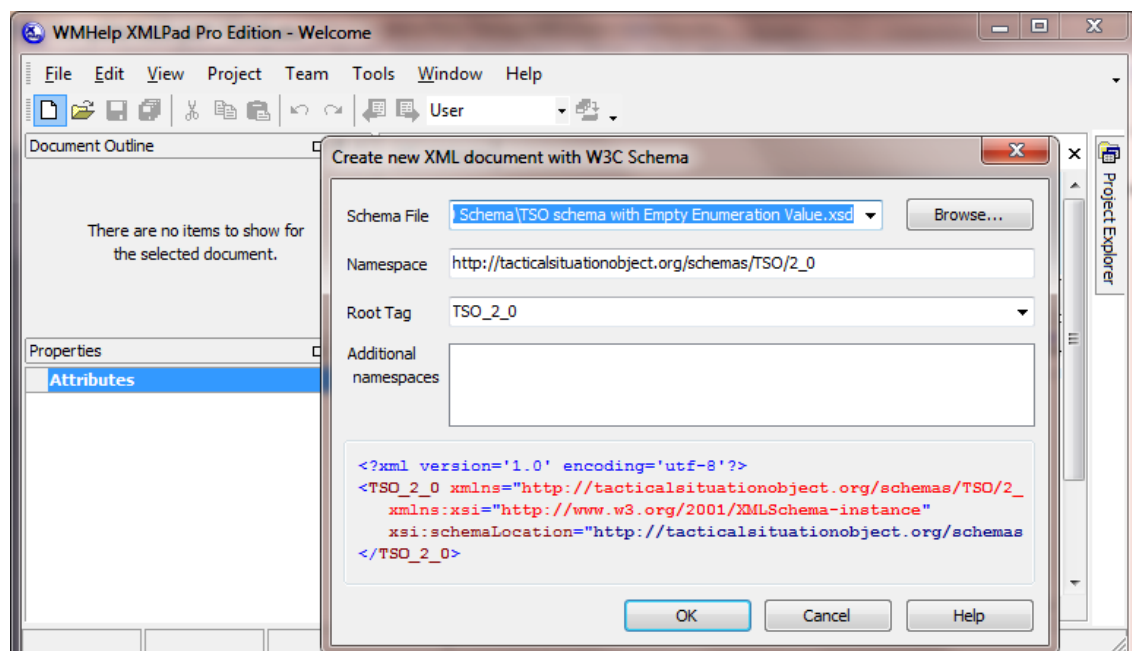


Figure 8.1: Specifying the TSO Schema for Constructing a TSO



In the first step of creating a new XML file using a schema, the XML editor asks for specifying the schema (Figure 8.1). When the TSO schema is specified, the (new) file opens including the root elements of TSO automatically as shown in Figure 8.2. The view of the document may be changed by clicking on ‘Grid View’ (some of the elements are highlighted with yellow shapes around them in Figure 8.3). When the right button of the mouse is clicked on the root element (TSO\_2\_0) of the XML file, a pop-up menu appears allowing to add CONTEXT, the first compulsory element of TSO (Figure 8.4). When the element (CONTEXT) is selected, it gets added to the document along with its compulsory child elements (ID, MODE and MSGTYPE as shown in Figure 8.5).

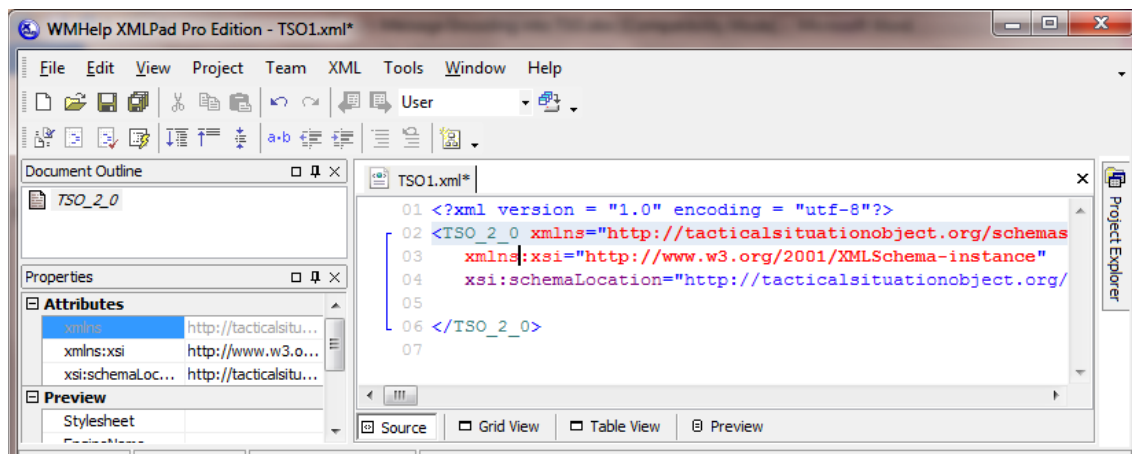


Figure 8.2: An XML file containing only the Root Element of the TSO

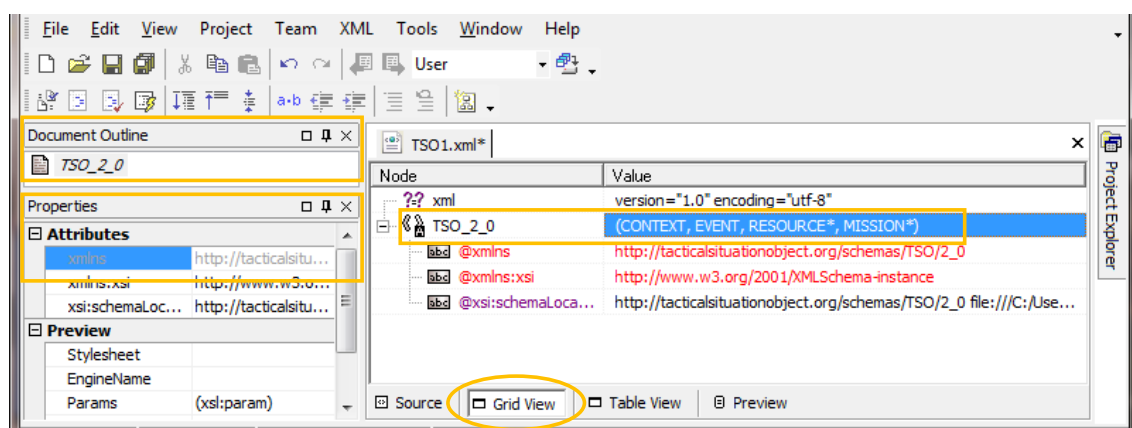


Figure 8.3: Creating TSO in Grid View

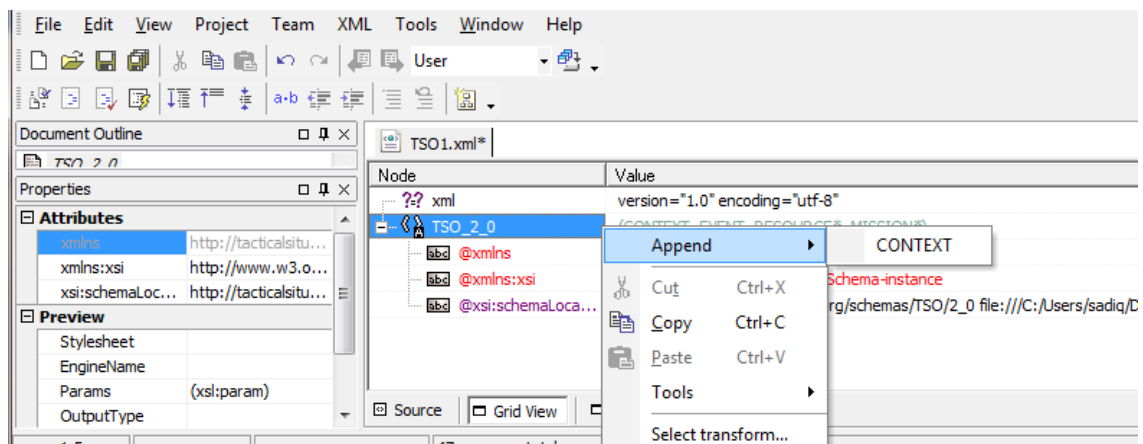


Figure 8.4: Using the Context Menu to Append a TSO Element

When the user clicks on a child element (e.g. ID and MODE) that accepts a value, the property pane (window) on the left provides either a text box for typing a value in or a drop-down menu for selecting one of the specified values depending on how the element is defined in the schema (Figure 8.6). Thus, a TSO can be easily constructed by repeating the steps shown for the rest of the TSO elements. How a complete TSO looks has already been shown in Figure 6.1 and Figure 6.2.

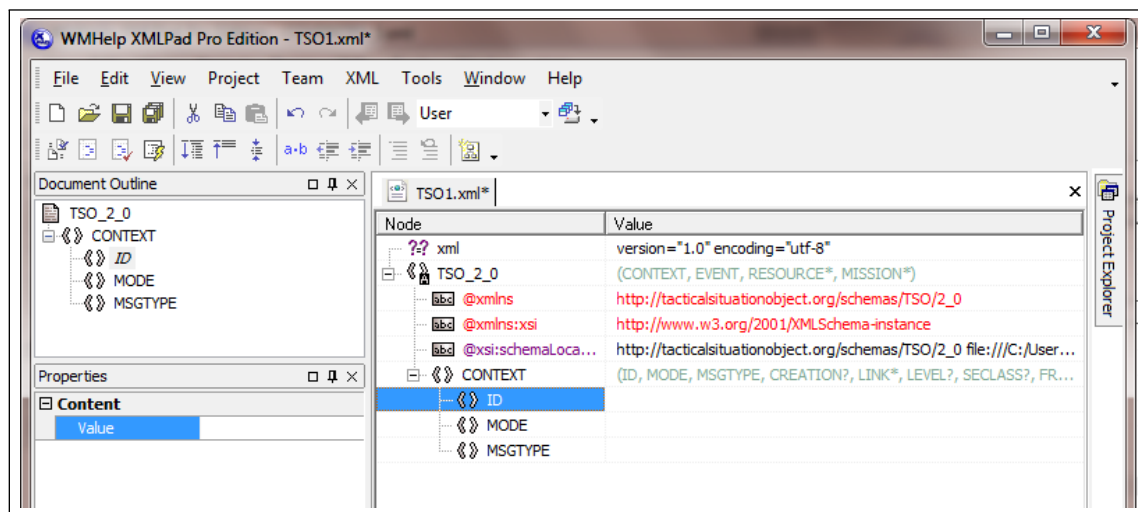


Figure 8.5: Compulsory Child-Elements of a TSO Element are Added Automatically

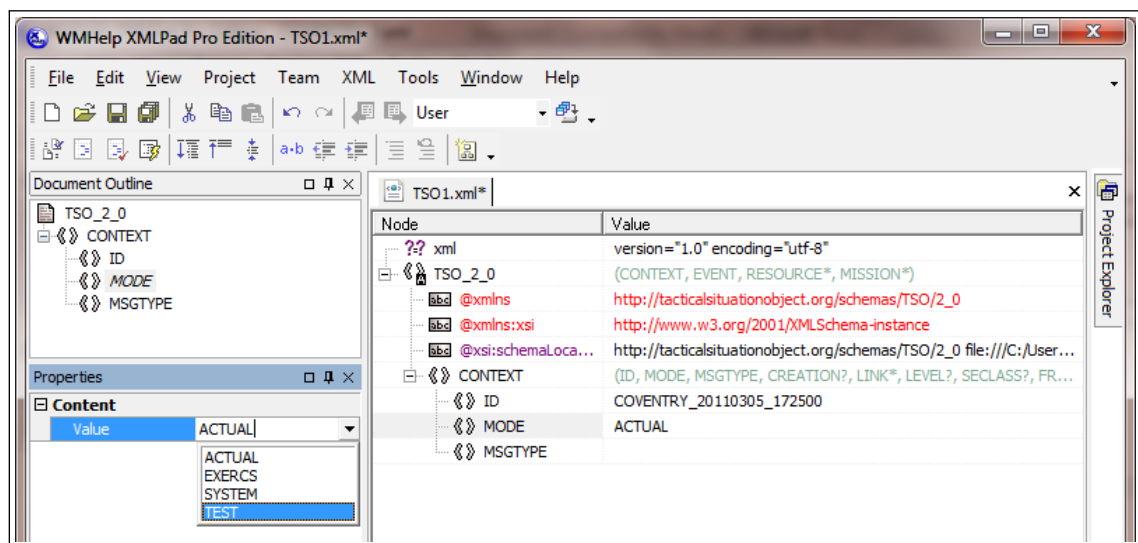


Figure 8.6: Context Aware Drop-down Menu Makes it Easy to Enter the Correct Values for TSO Elements

It is noteworthy that creating a complete TSO schema, which facilitates the construction of TSO, was a challenging task. While automatically modifying and extending an XML schema by adding new elements from a well-structured database is not a trivial task, the task of automatically creating the complete TSO schema was significantly harder due to the fact that the original (incomplete) TSO schema and the associated Data Dictionary had to be extracted from two PDF files that were their only available sources provided by the CEN (European Committee for Standardization). Hence, I had to accomplish the task by following multiple long, tedious and error-prone conversion methods and writing an application tool.

Although the complete schema that I constructed proved to be a very useful tool for constructing TSO, the process of encoding messages into TSO still remained a manual process due to the lack of the ability of understanding human natural language automatically. This remains as one of the future research topics.

## 8.2 Viewing TSO Data and Scoring Messages based on an Organisational Policy

The TSO Scoring System reads each TSO and assigns a score for Location of Informer, Freshness of Information and Reputation based on the organisational scoring policy defined in the database as shown in Figure 8.8. This application, developed in C#, uses the XML document object model for reading the content of each TSO.

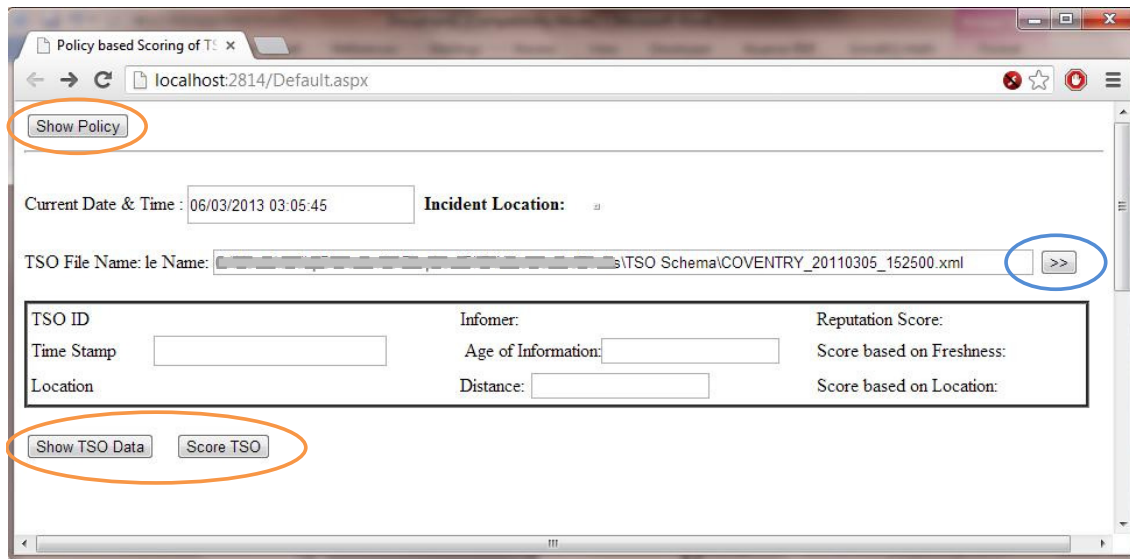


Figure 8.7: TSO Scoring System User Interface (View-1)

The application uses Google Geocoding API to retrieve the geographic coordinates (geocode) of a given address, postcode, street name, or public places like station and shopping centre. After having the geocodes of Incident Location and Informer Location, the application uses Google Maps JavaScript API V3 to show the locations on the map and to get the distance between the places. The application then calculates a score according to the organisational policy based on the informer's distance from the incident location and assigns the score to the respective TSO.

It is noteworthy that the user interface of the application is developed for demonstration purpose only, otherwise some of its features (such as adding, editing and deleting the scoring policy tables) would not be accessible to the same user who uses the system to score the messages and the scoring of the messages would not have been performed interactively for one message at a time. The scoring policy used in this system is not empirical either; it is an example only.

Figure 8.7 shows the user interface of the application. By clicking on the 'Show Policy' button, the user can view the policy tables that define what will the score of a message corresponding to Freshness of Information, Location of Informer, and the Reputation of Informer. When the policy tables are visible, as shown in Figure 8.8, the user can hide them by clicking on the 'Hide Policy' button. When the user clicks on the command button 'Show TSO Data' (as shown in Figure 8.7), the application performs the following actions:

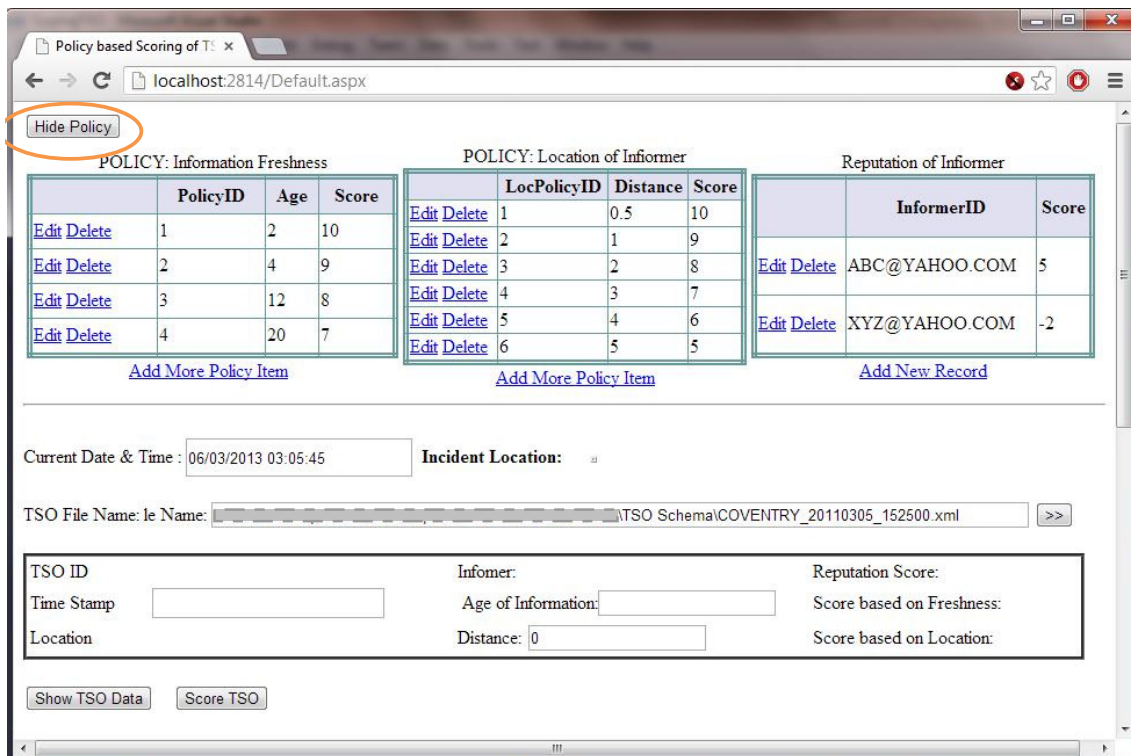


Figure 8.8: TSO Scoring System User Interface (View-2)

1. Reads the TSO corresponding to the file name shown in the text box labelled with 'TSO File Name'
2. Shows the information read from the TSO (e.g. TSO ID, Time Stamp, etc.) in the area surrounded with a black border (see Figure 8.9)
3. Works out the age of information by calculating the difference between the date and time in the TSO time-stamp and the current date and time
4. Shows the location of the associated incident and the informer's location on Google Maps (with a black and a red balloon markers respectively) while the map is centred at the incident location
5. Calculates and displays the distance between the incident location and informer location using the Google Maps service

When the user clicks on the 'Score TSO' button, the application also calculates the scores according to the policy shown in policy tables and assigns the scores to the TSO (as shown in Figure 8.10). The scored messages are also stored in an SQL

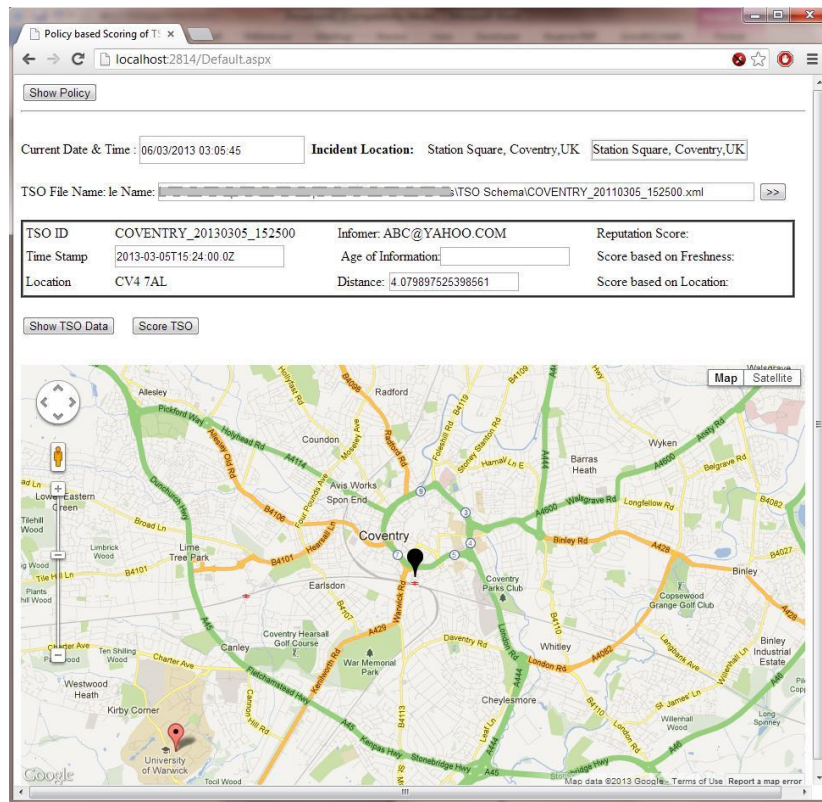


Figure 8.9: TSO Scoring System User Interface (View-3)

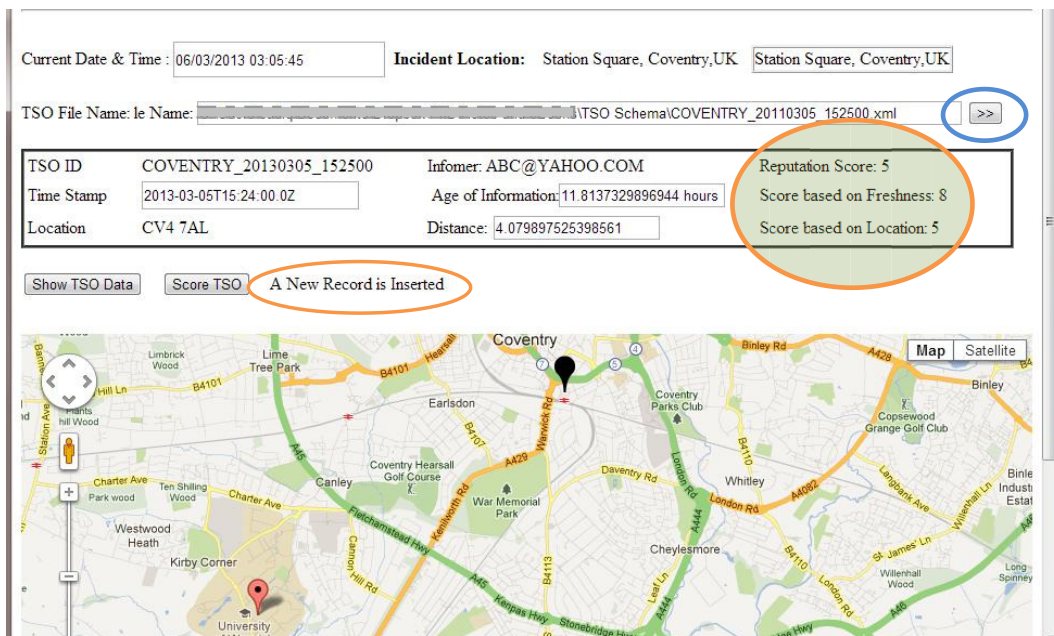


Figure 8.10: TSO Scoring System User Interface (View-4)

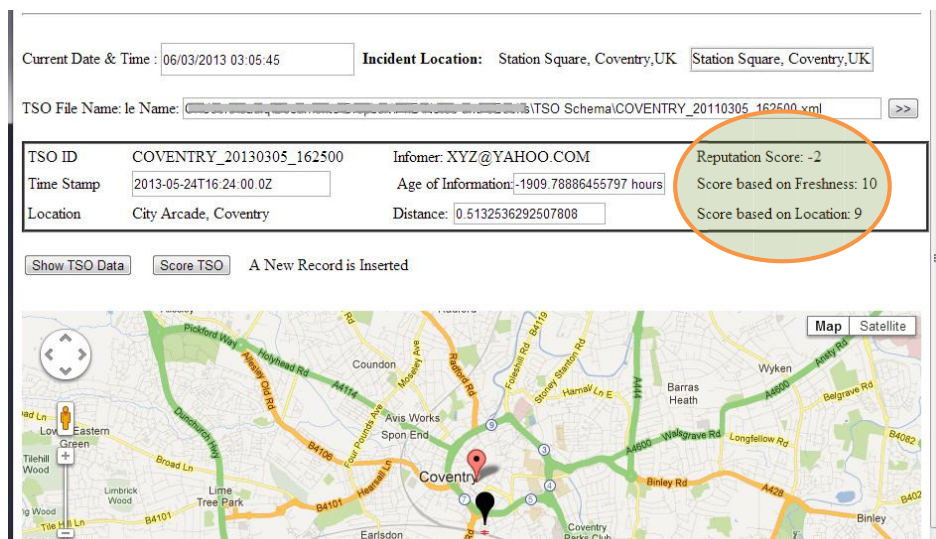


Figure 8.11: TSO Scoring System User Interface (View-5)

Server database table for further processing. After finishing scoring a TSO, the user can navigate to the next TSO by clicking on the button shown in Figure 8.7 with a blue circle around it. Figure 8.11 shows the data and scores associated with a different TSO.

### 8.3 Encoding TSO into Alloy and Feeding into Alloy Analyzer

As the structure and different components of an Alloy model have been described in detail in Chapter 7, recall that the building blocks of an Alloy model are Signatures (sets), Facts (constraints), and Predicates (expressions). Each Signature (labelled with the keyword **sig**) represents a set of objects. Facts and predicates are labelled with **fact** and **pred** respectively. Alloy models written in Alloy specification language are saved in files that are similar to normal text files and saved with an extension name '**als**' (e.g. Example.als). The Alloy Analyzer takes its input from the file *input.als*, for example, which can contain arbitrary number of messages. In a real world situation, a new predicate and other necessary constraints will be added to the model when a new message (TSO) will arrive. Since I am using a contrived and existing set of messages, all of the predicates are pre-written in an .als file. The Alloy model corresponding to the (received) messages are currently constructed manually. However, this can, in principle, be (largely) automated.

## 8.4 Construction of Consistent World Views through Alloy Automation

This section discusses the implementation of the ‘unit 5’ and ‘unit 6’ on the system architecture as shown in Figure 7.1. The Alloy Analyzer has been integrated with the Decision Support System (DSS) in order to perform the consistency analysis automatically and construct the world views. It is necessary to ensure a two-way communication between the Alloy Analyzer and the rest of the components of the DSS, so that a message or a cluster of messages can be sent to Alloy Analyzer for processing and the result returned by the Analyzer can be retrieved. However, Alloy Analyzer is developed by MIT, written in Java and distributed as a JAR (Java ARchive) file, while the rest of the components of the DSS are developed in C# (by me). Hence, it is not a trivial task to establish a two way communication in an elegant way for a few reasons:

1. Input has to be given in the right way and in the correct format expected by the API. Although Alloy Analyzer reads the input data (messages/predicates) from a (.als) file, it expects other information (input file’s name and location) to be passed to it as input parameters.
2. The invoking function needs to know whether Alloy Analyzer has finished its operations and is ready to take more input for further processing.
3. Output returned by Alloy Analyzer, after procession a given input, determines what the next input should be.
4. Logical reasoning is known to be a time consuming operation. Hence, applications developed for this purpose should avoid using any slow and inefficient processes.

In order to overcome the issues outlined above, I have written a small web service in Java which serves as a wrapper for the Alloy Analyzer. The web service receives requests from the DSS and invokes the Alloy Analyzer with necessary input parameters (provided by the invoking function of the DSS) and makes the output (returned by the Alloy Analyzer) available to the invoking function of the DSS.

Initially, I wrote a different wrapper function (in Java) that used to invoke the Alloy Analyzer in response to the request received from one of the components of the DSS, which is written in C#. The main difference between the initial wrapper



function and the current web service version of the wrapper is that it used to write the output returned by the Alloy Analyzer into the system console, so that the invoking component of the DSS could read that output from the console. However, it turned out to be a slow process. Hence, I have replaced it with the current wrapper which is technically a web service.

In the DSS application, Alloy is used mainly in two steps. In the first step, the DSS application invokes Alloy Analyzer to check the satisfiability of all individual messages from the input file. It may be noted that a valid (syntactically correct) input file may contain one or more predicates and Alloy Analyzer can check the consistency of all of the predicates at once, or it can selectively check the consistency or satisfiability of any specific predicate using the ‘run’ command. In the second step, the application automatically creates an input file (from a pre-built template) by generating multiple clusters of messages (one cluster at a time) with different combinations of individual messages (in an attempt to form world views with the largest consistent set of messages described in Section 7.4.1).

The template file contains the complete model including all predicates and an incomplete definition of a compound predicate. Figure 8.12 shows an overly simplified structure of the template file, which has been provided only to illustrate how the (incomplete) compound predicate looks like. In Figure 8.12 the incomplete compound predicate is named as *CompositePredicate*. The incomplete predicate in the template is later updated programmatically with the necessary statements as illustrated in Figure 8.13. It may be noted that the overly simplified template illustrated in Figure 8.12 does not include the proper definitions of signatures and predicates, and the associated facts or restrictions. Hence, the inter dependency of the signatures and predicates are not clearly visible. For example, none of the predicates shown in the template file take ‘person’ as an input parameter although there has to be some ‘person’s in the universe of discourse for the predicates to satisfy.

```

sig Event
{
    .....
}

sig Person
{
    .....
}

fact
{
    .....
}

pred Example_1 [roadAccident:Collision, car1, car2: Vehicle]
{
    .....
}

pred Example_2 [roadAccident:TransportIncident, car1: Vehicle]
{
    .....
}

pred Example_3 [event: SendingMessage]
{
    .....
}

pred Example_4 [roadAccident:TransportIncident]
{
    .....
}

//Check the Satisfiability of the Predicate
run CompositePredicate for 4 but 7 int, 3 Car, 2 Person

pred CompositePredicate[thisEvent:Event, car1, car2, car3 : Vehicle]
{
    //Content of this predicate will be written programatically
}

```

Figure 8.12: A Sample and Incomplete Alloy Input File

```

.....

pred Example_1 [roadAccident:Collision, car1, car2: Vehicle]
{
    .....
}

pred Example_2 [roadAccident:TransportIncident, car1: Vehicle]
{
    .....
}

pred Example_3 [event: SendingMessage]
{
    .....
}

pred Example_4 [roadAccident:TransportIncident]
{
    .....
}

//Check the Satisfiability of the Predicate
run CompositePredicate for 4 but 7 int, 3 Car, 2 Person

pred CompositePredicate[thisEvent:Event, car1, car2, car3 : Vehicle]
{
    //Content of this predicate is written programmatically

    Example_1 [thisEvent, car1, car2] &&
    Example_2 [thisEvent, car3] &&
    Example_3 [thisEvent] &&
    Example_4 [thisEvent]
}

```

Figure 8.13: A Sample Alloy Input File after Updating it Programmatically

Each auto-generated input file containing a compound predicate is then fed to the Alloy Analyzer for satisfiability checking. Alloy Analyzer returns true or false depending on the satisfiability of the messages or message clusters and the DSS application stores the returned values in database as shown in Figure 7.12 and Table 8.2 as an example. The second step is repeated many times to ensure that the

application generates the complete set of possible world views. The DSS application also ensures that the generated world views are all unique (distinct) and none of them are a subset of another larger world view.

Table 8.2: Message Clusters and World Views in Database

Message-Cluster ID	Satisfiable	World View	Score
<b>AFGIJ</b>	<b>TRUE</b>	<b>TRUE</b>	<b>14.52</b>
<b>ABDI</b>	<b>TRUE</b>	<b>TRUE</b>	<b>14.64</b>
<b>AEI</b>	<b>TRUE</b>	<b>TRUE</b>	<b>16.14</b>
<b>AHI</b>	<b>TRUE</b>	<b>TRUE</b>	<b>14.43</b>
AFGI	TRUE	FALSE	0
ABD	TRUE	FALSE	0
AFG	TRUE	FALSE	0
AIJ	TRUE	FALSE	0
AB	TRUE	FALSE	0
AE	TRUE	FALSE	0
ABDIJ	FALSE	NULL	0
ABDE	FALSE	NULL	0
ABDF	FALSE	NULL	0

Note: Message clusters **AFGI**, **ABD**, **AFG**, **AIJ**, **AB**, and **AE** are all satisfiable; yet their value in the third column is 'False' i.e. they are not world views. This is because each of them is a subset of another larger cluster which formed a world view e.g. ABD is a subset of ABDI.

In the last three rows, the message clusters (**ABDIJ**, **ABDE** and **ABDF**) have the value 'False' in the second column, i.e. they are not satisfiable. Hence, their value in the 'World View' column is NULL; which means it has not been checked whether they form any world view, as it is unnecessary.

The implementation of the decision support system (DSS) includes two database tables (SQL Server). One stores the names of the predicates corresponding to each TSO (message) and their truth values returned by Alloy Analyzer (as shown in Figure 7.12). The other table stores compound predicates (cluster of messages constructed following the procedure described in Section 7.4.1) and their truth values.

## 8.5 User Interface: Presenting an Ordered List of World Views for Decision Makers

The unordered world views created (and saved in the database) by the consistency checking unit (unit 5 on the system architecture in Figure 7.1) are presented in

front of the decision makers by ordering them according to their total score. The visualisation of the world views has been performed by using a Bubble chart, in which each bubble represents a world view. The size of the bubbles (i.e. the world views) and their relative position on the chart indicates their score. When the user hovers the mouse pointer over a world view, its total score and the number of messages contained in that world view are displayed on the chart, as shown in Figure 8.14.

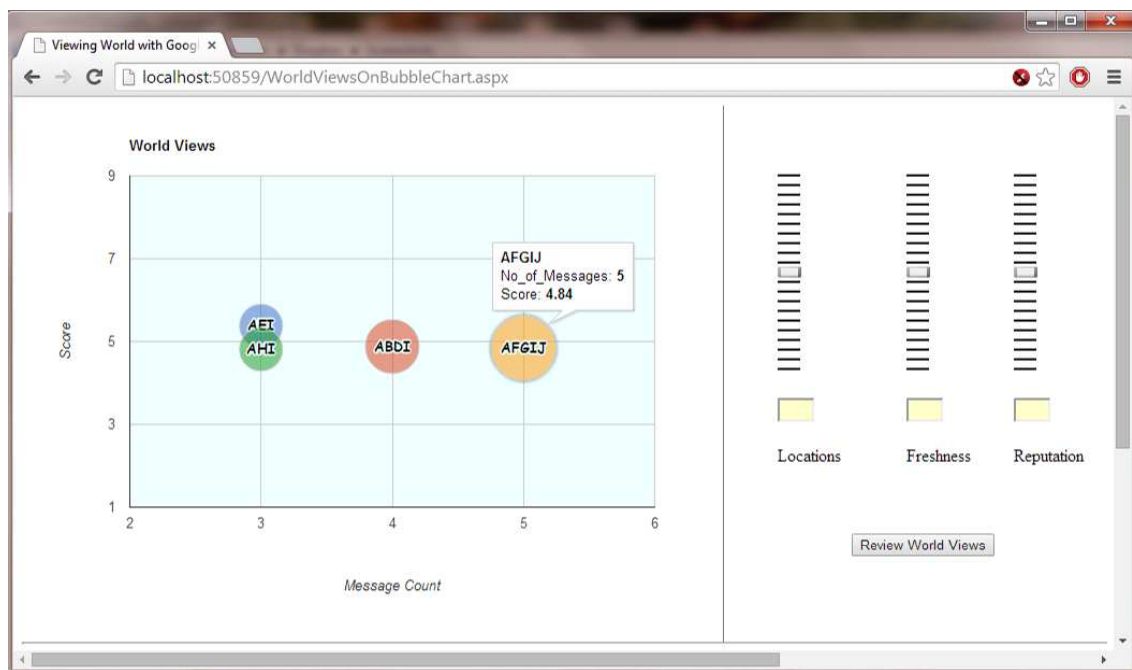


Figure 8.14: World Views Before Applying Decision Making Policy

When the user clicks on any of the world views (bubbles), the messages included in the selected world view are also displayed in a separate table below the chart, as shown in Figure 8.15. Presenting the world views in the form of bubbles makes it easy for the system to present a large volume of information in an intuitive way. The purpose of using the Bubble chart to represent the world views is that it enables the system to present a large volume of data in a simple and intuitive way. This will also save the user from information overloading by enabling them to view the information belonging to only that world view, in which the user is interested.

The Bubble chart is implemented using Google Chart Tools<sup>1</sup> as Google provides APIs (Application Programming Interface) for different types of visualisation tools.

<sup>1</sup><https://developers.google.com/chart/>

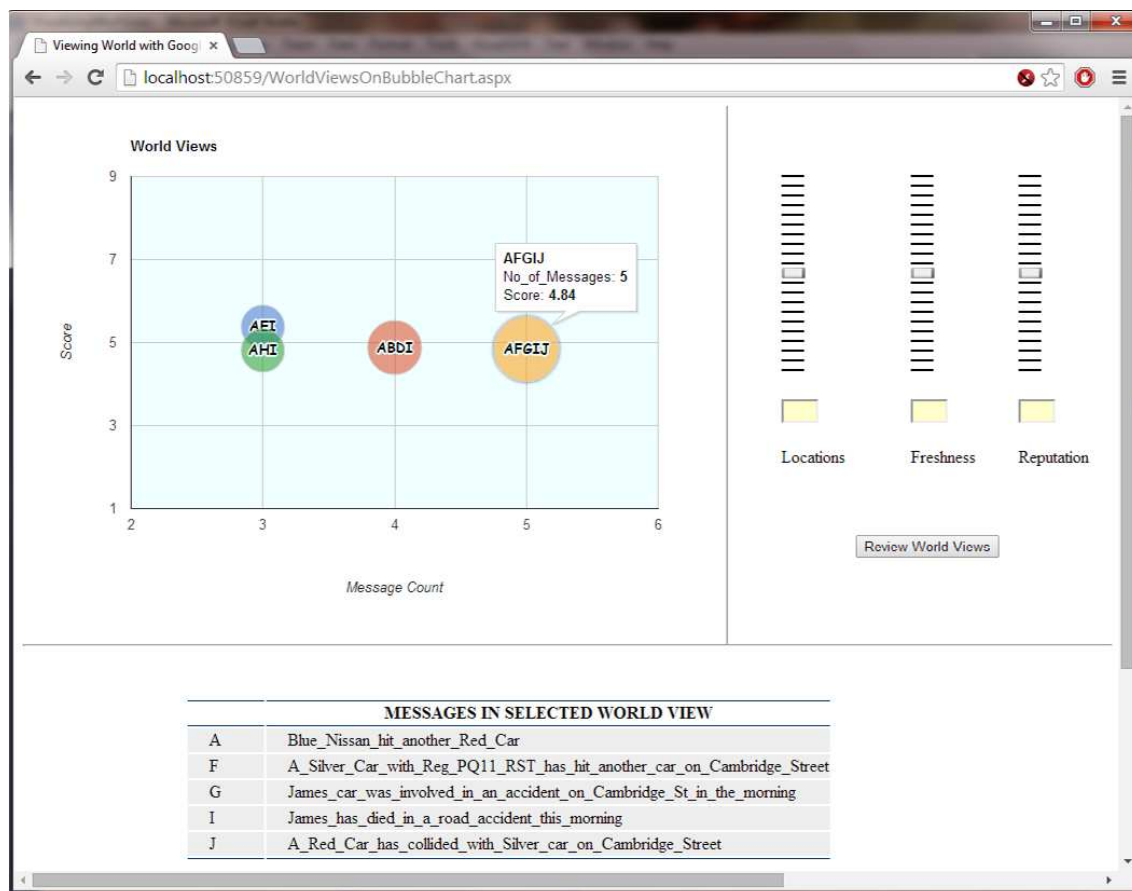


Figure 8.15: Messages in a Selected World View

Google Charts are based on JavaScript and requires the input data to be in JSON (JavaScript Object Notation) format. However, since JavaScript is a client-side scripting language and our application is based on ASP.Net, a server-side Web application framework, data synchronisation between JavaScript (on client-side) and ASP.Net (on server-side) is a bit of challenge. Because, when the user selects a world view by clicking on the Bubble chart, the JavaScript handles the user interaction and responds accordingly although the ASP.Net based main part of the application remains unaware of the user's selection. Therefore, it requires further actions to make the user input available to the server, so that the server can provide the necessary information associated with the selected world view such as the score of the world view, the number of messages contained it, etc.

### 8.5.1 Synchronising Client-side and Server-side Data

The application reads the details of each world view from the database and feeds the data into the Google Chart for generating a Bubble chart representing the world views. Data read from the database is converted into JSON (JavaScript Object Notation) format and then stored in an ASP *HiddenField* control to make the data available to the client-side scripts (JavaScript). JavaScript reads the input data from the hidden field and generate the world view chart using Google Chart API. When a user clicks on a world (Bubble), JavaScript captures the selected world's ID and stores it in an invisible *TextBox* control because the server script needs this information to query the database and retrieve all messages included in the selected world-view.

#### Querying the Database for the Selected World View Data

Querying the database using the selected world's ID is implemented in an event-driven procedure that executes every time a different world view (bubble) is selected (by the user) i.e. the content of the invisible *TextBox* changes and its *ontextchanged* event occurs. However, the problem is that since the world views (Google Chart) are not generated using ASP controls, server-side script (ASP) does not know anything when a user clicks on a world view although the JavaScript is aware about this click event and its related data. To leverage this gap between the client-side and server-side scripts, an artificially post-back is generated using JavaScript by invoking the function `__doPostBack('txtSelectedWorld')`, which is automatically generated by the ASP.NET engine and included in the final HTML page. Therefore, the `__doPostBack(eventTarget, eventArgument)` function does not exist until the requested web page is loaded on the browser. Since the function `__doPostBack` is called with the parameter 'txtHiddenControl', the postback appears to be caused by the invisible *TextBox* control named as *txtHiddenControl*. It needs to be noted that ASP.NET engine does not include the `__doPostBack(eventTarget, eventArgument)` function in the HTML code of the resultant webpage unless there is at least one ASP.Net control on the page that has its *AutoPostBack* property set to 'True' i.e. `AutoPostBack="True"`. This is why an invisible *TextBox* control has been used to stored the ID of the user-selected world view in stead of a *HiddenField* control. The difference between a *HiddenField* control and an invisible *TextBox* control is that the *TextBox* control has an *AutoPostBack* property which is necessary to send

data from the client-side to the server, whereas the HiddenField control does not have a 'AutoPostBack' property. Therefore, the data stored in a HiddenField may remain inaccessible to the server-side controls if there is no other control on that page with its AutoPostBack property set to 'True'.

## 8.6 Application of Decision Making Policy

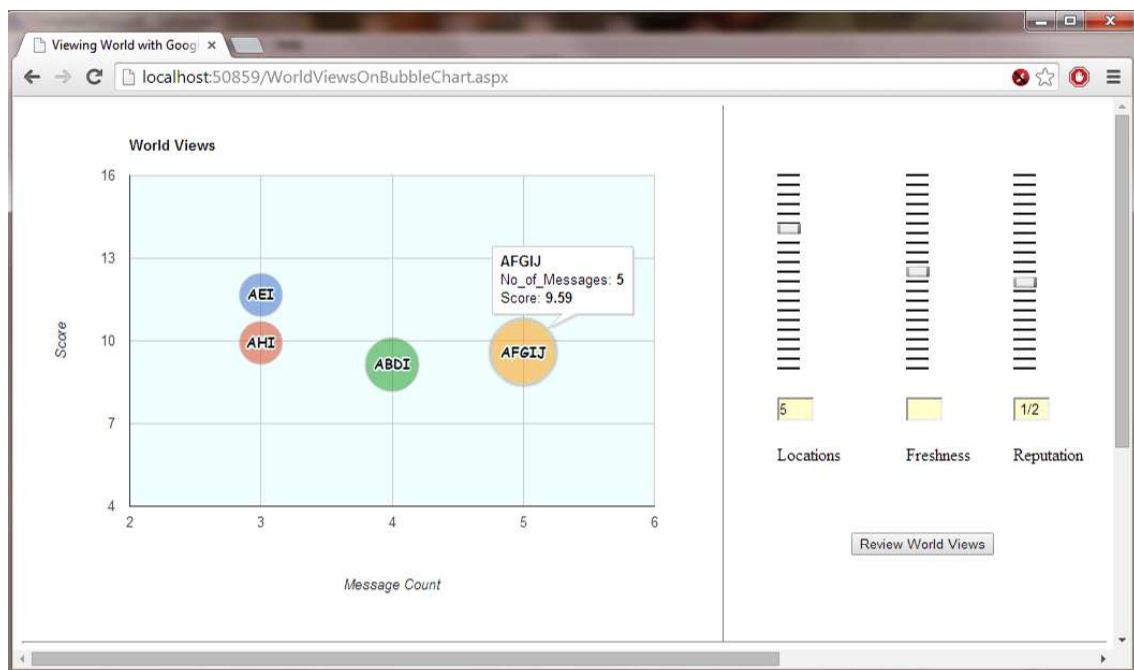


Figure 8.16: World Views After Applying Decision Making Policy

As the decision maker may want to amplify or attenuate some of the provenance and trust factors, the user interface of the system includes a control panel consisting of some controls that look similar to graphic equalisers as shown on the right side of the word views in Figure 8.14. The selected value using a slider control is shown in a box below each slider. The slider controls and the corresponding Textboxes are synchronised with AJAX (Asynchronous JavaScript and XML) technology. Therefore, the values in the Textboxes change without requiring the whole page to reload. The decision maker applies different weights to different factors by clicking on the 'Review World View' button after selecting the desired values using the sliders. Figure 8.16 shows that the world views have changed after applying the decision making policy on the world views shown in Figure 8.14. The effect of Decision Making Policy can be understood better in Chapter 9, which presents the



scores of each individual message, scores of the world views constructed from those messages and how their scores changed after the policy is applied.

Although AJAX technology has been used in the implementation of this component, the sizes and order of the world views do not change as the user moves the sliders up and down. It only changes the values of the accentuation and attenuation factors in the boxes below the sliders. The user is required to click on the button Review World Views to see how the changes affect the size and order (relative position) of the world views. This feature needs to be improved so that the use of the AJAX technology becomes more evident and the system becomes more user-friendly.

## 8.7 Lines of Code Metrics

I have provided the metrics on the Logical Lines of Code (LLOC) that I have written to develop the Decision Support System (DSS) in Table 8.3. The Logical Lines of Code excludes the comments, blank lines and extra lines used in multi-line statements. Table 8.3 does not provide any account for the HTML and CSS (Cascading Style Sheets) codes, as they are mostly (though not all) auto-generated. The code metrics will provide an estimation of how much effort may be necessary to reproduce the system. The following utility tools and services have been used to count the Lines of Code:

- *Jsmeter*<sup>2</sup>: It is an open-source service for counting Lines of Code (LOC) in JavaScripts.
- *Locmetrics*<sup>3</sup>: I have used this utility tool for counting the Lines of Code written in SQL and Java, although it is capable of counting LOC for other languages too.
- *Code Metrics Viewer 2010*<sup>4</sup>: Code Metrics Viewer is based on Microsoft's Visual Studio Code Metrics Powertool 10.0<sup>5</sup>, a command line utility to calculate code metrics for .NET code. Code Metrics Powertool counts the lines of code from the Common Intermediate Language (CIL) code, which is produced after compiling the source code. Hence, the number of lines of code counted by

---

<sup>2</sup><http://jsmeter.info/>

<sup>3</sup>[www.locmetrics.com/](http://www.locmetrics.com/)

<sup>4</sup><https://visualstudiogallery.msdn.microsoft.com/9f35524b-a784-4dbc-bd7b-6babd7a5a3b3>

<sup>5</sup><http://www.microsoft.com/en-us/download/details.aspx?id=9422>

Code Metrics Powertool is not the exact number of lines in the source code file; it provides an approximate number of lines of code as the lines are counted from the Intermediate Language code [93].

Table 8.3: Metrics on the Lines of Code written to develop the DSS Framework

Application Modules	Logical Lines of Code			
	JavaScript	Java	C#	SQL
Harvesting Open-source Data - Collecting Tweets from Targeted User Accounts	-	-	140	-
Harvesting Open-source Data - Collecting Tweets using Twitter Search API	-	-	322	-
Reproducing a Graphical Timeline of Incident-related Tweets	42	-	98	-
Tool Support: Finding Popular Users by Analysing Retweets and Mentions	-	-	217	-
Constructing TSO Schema from TSO Message Structure and its Data Dictionary	-	-	323	-
Viewing TSO Data and Scoring Messages based on an Organisational Policy	39	-	223	-
Clustering TSO based on Event Location, Event Type, Actors and Sentiment/Positivity	-	-	591	-
World View Generator and Interface to Alloy Analyzer	-	-	758	-
Web Service Wrapper for the Alloy Analyzer	-	36	-	-
User Interface: Presenting an Ordered List of World Views for Decision Makers	73	-	111	-
SQL Functions and Stored Procedures	-	-	-	138
<b>Total Lines of Code =</b>	<b>154</b>	<b>36</b>	<b>2783</b>	<b>138</b>

## 8.8 Summary

The implementation of the decision support system and its usage have been described in this chapter. The system facilitates encoding messages into TSO, scoring the messages, consistency analysis of the messages, constructing world views and applying the decision making policy. Since the system has been developed as a prototype, some details have been omitted intentionally. For example, only three of the provenance and information quality factors (Location, Freshness and Reputation) have been used in the prototype.

The implemented system is predominantly a web application written in C# using the ASP.NET framework. However, some of its components are written in Java and JavaScript as the system uses a number of third-party tools and services that are written in other languages including Java. One of the main third party tool integrated with the system is the Alloy Analyzer, a Java based model finding application developed at MIT, which performs the logical consistency analysis. The other third party services used in the system are Google Maps, Google Geocoding Service, and Google Charts. The input and output data of the system are mostly saved in an SQL Server database, although some data is saved in other different formats (e.g. XML, JSON) too, where they are most appropriate. The Lines of Code metrics corresponding to different components of the DSS have been provided to give an estimation of how much effort may be necessary to reproduce the system.

The entire decision support system is automatic except the creation of TSO and encoding messages into Alloy. However, modelling the messages into Alloy can, in principle, be (largely) automated, although encoding the messages into TSO may be possible only through natural language processing (NLP). Since, exploring the possibility of using NLP is beyond the scope of this thesis, it has been left as a topic for future research.

## CHAPTER 9

# Technical Demonstration of the Decision Support System

The previous chapters have described the motivation behind the development of the DSS, the design principles, operation procedure and implementation of the system. In this chapter, I will demonstrate step by step, how the user can use the system using some contrived data. This chapter also presents some experiments that I have carried out with some test data, in order to see whether the system performs the consistency checking and other desired operations correctly and produces expected results. The experiment result demonstrates some desired properties of the decision support system and the world views. The result also illustrates the effectiveness of the system and supports the underlying theory or concept.

Here is a list of questions and some interesting phenomena of the world views that the experiment results are expected to answer or demonstrate:

1. Is the consistency evaluation of each individual message performed by the DSS sound?

The soundness of the system will be proved if –

- (a) None of the messages that are classified as consistent by the system is in contradiction with any of the facts.
  - (b) Each of the messages that are classified as inconsistent by the system is in contradiction with at least one of the facts.
2. Does the DSS generate the complete set of the largest possible world views from the available messages? Are the world views generated by the DSS logically sound?

If the DSS generates the complete set of the largest possible world views, then none of the world views can be any bigger and there is no such cluster of messages that is a world view but the DSS does not identify it as a world view.

3. Every new message makes a change to the existing world views.

It has been stated and proved in Section 7.5 that although the number of world views may vary due to an increase in the number of input messages, there is no guarantee that the number of the world views will increase. However, it is certain that the number of world views will not decrease in any case, with an increase in the number of messages. A new message will leave either of the two effects:

- It will be included in one or more existing world views and (thus,) it will make those world views richer by being supportive of them
- It will increase the number of world views by forming new world views together with other messages or on its own.

To prove that the DSS is functioning correctly, its output must show the same property.

4. How can the Decision Making Policy affect the world views?

If the Decision Making Policy is applied then the perceived reliability of world views may change. An otherwise more reliable world view may become less reliable and vice versa.

5. If a premise is satisfiable, then its consequent is satisfiable too.

If a message ( $M'$ ) is a logical implication (entailment) of another message  $M$  i.e  $M \rightarrow M'$ , then any world view which includes  $M$  will also include  $M'$  in it. In other words, there must not be any world view that includes  $M$  but not  $M'$  when  $M \rightarrow M'$ . However,  $M'$  may be included in a world view which does not include  $M$ .

6. How fast is Alloy and Alloy Analyzer? Can Alloy be used to evaluate the consistency of reasonably large number of messages possibly in a real life incident?

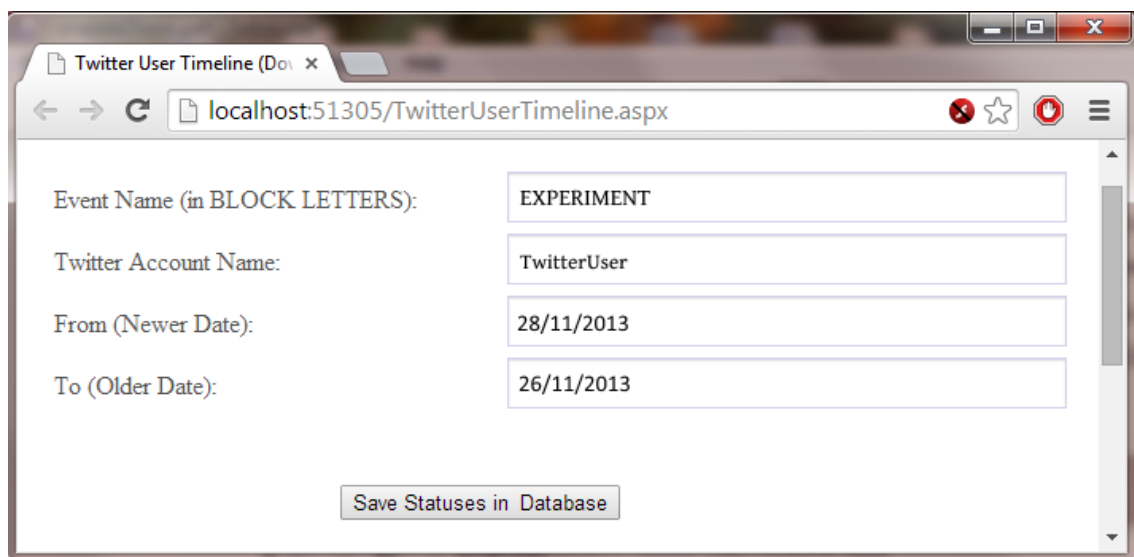
Overall, this chapter shows how to use the DSS and presents the experiment in detail along with the test data and the result of the experiment. The experimentation has been carried out with a very small set of data to make the automation (automatic experimentation) feasible.

## 9.1 How to Use the System

In the previous chapters, we have learned about the Decision Support System framework and how the DSS is implemented. This section (and its subsections) demonstrates the application of the whole framework and how the user can use the system, step by step. Table 8.1 (in Chapter 8) will be very useful in following this step by step demonstration, as it has outlined the interfaces between different components (i.e. what the inputs and outputs of each of the components are and their data format) of the DSS.

### 9.1.1 Harvesting Data

How data can be harvested from open sources, more specifically from Twitter, has already been described in Chapter 3. Chapter 3 also describes how the data harvesting applications/tools work. The only thing that user needs to do is filling up the forms (shown in Figure 9.1 and Figure 9.2) with appropriate parameters depending on their needs. The harvested data gets stored in the database automatically.



The image shows a web browser window with the following content:

- Browser title: Twitter User Timeline (Do... x)
- Address bar: localhost:51305/TwitterUserTimeline.aspx
- Form fields:
  - Event Name (in BLOCK LETTERS): EXPERIMENT
  - Twitter Account Name: TwitterUser
  - From (Newer Date): 28/11/2013
  - To (Older Date): 26/11/2013
- Button: Save Statuses in Database

Figure 9.1: GUI for Harvesting data from a User's Timeline on Twitter

Figure 9.2: GUI for Harvesting data using Twitter Search API

### 9.1.2 Data Set

The tweets returned by the Twitter APIs contain the messages written in plain text, as well as other metadata (e.g. time-stamp, user-name, user-location) as shown in Section 3.1.1 and Appendix A. Suppose the user has collected the ten messages shown in Table 9.1. The table (Table 9.1) also shows the time when each of the messages were created by the Twitter users (informants) and the locations of the Twitter users. As outlined in Table 8.1, the Twitter data comes in XML and JSON format and the plain text messages along with other metadata contained in tweets are stored in an SQL Server Database before transformed into TSO.

Table 9.1: Test Data Set-1: The Messages Used to Construct World Views and their Satisfiability Evaluated by the DSS

Msg ID	Message	Date & Time	User's Location
A	Blue Nissan hit another Red Car	05/03/2011 02:21:01	Location-A

*Continued on next page*

Table 9.1 – Continued from previous page

<b>Msg ID</b>	<b>Message</b>	<b>Date &amp; Time</b>	<b>User's Location</b>
B	Car with Reg. No. AB11_XYZ hit a Red BMW on Oxford Street	03/03/2011 20:21:01	Location-B
C	Green Car with Reg AB11_XYZ hit Red BMW on Oxford Street	05/03/2011 12:21:01	Location-C
D	Blue Car with Reg. No. AB11_XYZ hit a Red Car on Oxford Street	04/03/2011 20:21:01	Location-D
E	Blue Car with Reg. No. AB11_XYZ hit a Red Car in Oxford	05/03/2011 06:21:01	Location-E
F	A Silver Car with Reg. No. PQ11_RST hit another car on Cambridge Street	04/03/2011 20:21:01	Location-F
G	James's car was involved in an accident on Cambridge St at 10:30 AM	05/03/2011 14:21:01	Location-G
H	James had an accident with his car in Cambridge at 10:30 AM	05/03/2011 06:21:01	Location-H
I	James has died in a road accident this morning at 10:30 AM	05/03/2011 16:21:01	Location-I
J	A Red Car has collided with a Silver car on Cambridge Street	05/03/2011 12:21:01	Location-J

### 9.1.3 Encoding of Plain Text Messages into TSO

The user now needs to encode these plain-text messages and other related information (metadata) into TSO for being able to automatically assign a score corresponding to each trust and provenance factor. The currently manual process of encoding messages into TSO using the TSO schema and any standard XML editor will yield TSOs. A TSO corresponding to a similar message is shown in Section 6.1.1, which demonstrates the mapping of data and metadata into different TSO Components.

### 9.1.4 Assigning Scores against Trust Factors

Suppose the location where an incident has taken place is *Station Square, Coventry* and the time when the user of the DSS was processing the harvested information



is 20:21:01 2013/03/05. Section 5.1.3.1 defines a scoring policy and demonstrates how the scoring should be accomplished. Section 8.2 also showed how the distance between two locations can be worked out using Google Maps and Geocoding services. By following the scoring method and examples given in Section 5.1.3.1 and Section 8.2, we can calculate the age of the messages (Time of Processing - Time when Message was Posted by the informant) and the distance between the incident location and informant's (Twitter User) location. We can also work out the scores corresponding to different trust factors for each of the messages, by following the scoring policy defined in Section 5.1.3.1. These scores are also stored in database. Suppose Table 9.2 shows all these information and scores that we have calculated. However, the reputation score can only be obtained from the user's history of reputation.

Table 9.2: Proximity of User, Age of Information and Scores

Msg ID	Proximity of User (Mile)	Age of Information (Hour)	Scores		
			Location	Freshness	Reputation
A	7	18	4	4	5
B	9	48	3	1	8
C	5	8	5	7	-2
D	5	24	5	3	5
E	3	14	7	5	3
F	9	24	3	3	2
G	5	6	5	8	4
H	7	14	4	5	1
I	4	4	6	9	5
J	7	8	4	7	3

### 9.1.5 Consistency Analysis, World View Generation and Scoring

After having the messages and associated scores calculated, the user now needs to perform the consistency analysis for constructing world views. Two more things need to be done prior to checking consistency of the harvested data:

1. Set out the facts and physics related the data set.
2. Specify the facts and messages in Alloy specification language.

The facts and assumptions that have been used to verify the consistency of messages are listed below.

- Fact - 1: At the time of an incident, the location of actors and casualties is the same as event location.
- Fact - 2: An object can be at a location for any period of time but it cannot be at two locations at the same moment in time.
- Fact - 3: A vehicle can have only one driver at a given time, although it can be driven by many drivers at different times.
- Fact - 4: A driver can drive the same vehicle many time.
- Fact - 5: Sometimes, there may be no driver in a vehicle at all<sup>1</sup>.
- Fact - 6: Two vehicles cannot be driven by the same driver at the same time.
- Fact - 7: Two vehicles cannot have same registration number.
- Fact - 8: At a given time, the location of a vehicle and its driver is same.
- Fact - 9: At least one of the vehicles involved in an accident has to have a driver in it. (However, in reality, a vehicle can hit another car/object, even if there is no one in the vehicle(s). For example, a car parked on a downhill slope may hit another object after losing its brakes.)
- Fact - 10: A vehicle has to have only one registration number.
- Fact - 11: A car can have only one colour.
- Fact - 12: The vehicle with registration number AB11\_XYZ is blue car (Nissan)<sup>2</sup>.
- Fact - 13: The vehicle with registration number PQ11\_RST is silver car (Volkswagen)<sup>2</sup>.
- Fact - 14: Blue is not Silver and Silver is not Blue. This is true for all colours.
- Fact - 15: The vehicle with registration number PQ11\_RST is owned by James<sup>2</sup> .
- Fact - 16: James does not allow anyone else to drive his car
- Fact - 17: If a person or an animal dies, then it is dead ever after.

---

<sup>1</sup>A vehicle can be both stationary and moving (on the street) without having a driver in it. For example, a vehicle, which was parked in a slope, can go downhill and hit another object by losing its handbrake (parking brake), while there is no driver in it, even though it is an ordinary (not unmanned) vehicle.)

<sup>2</sup> These assumptions are considered as facts because the *Vehicle Licensing Authority* (e.g. DVLA in the UK) holds all this information in their database

Fact - 18: A person or animal which is dead cannot perform any action (on its own).

Fact - 19: If an actor (man/animal) does an action, then s/he is not dead and s/he was not dead before.

Fact - 20: Day-1 comes before Day-2. The same rule applies to hour, minute and second.

Fact - 21: London, Oxford and Cambridge are three different cities (Location)

Fact - 22: Oxford Street and Cambridge Street are two different streets (Location) in London

As mentioned earlier, specifying the facts and messages in Alloy is currently a manual process. However, how the facts and messages can be specified in Alloy is described in Section 7.3. The Alloy specification of the facts and messages are shown in Appendix C and they are stored in a (text) file with extension ‘.als’ (e.g ‘Alloy.als’). How the consistency checking of the messages should be performed automatically is also described in Section 7.3. Table 9.3 shows the result of consistency checking of the messages (listed in Table 9.1). This is one of the outputs of the ‘Consistency Analysis and Conflict Resolution’ unit (Unit 5 on Figure 7.1). The ‘Consistency Analysis and Conflict Resolution’ unit stores all its output in an SQL Server database.

Table 9.3: Test Data Set-1: The Messages Used to Construct World Views and their Satisfiability Evaluated by the DSS

<b>ID</b>	<b>Proposition</b>	<b>Satisfiable</b>
A	Blue Nissan hit another Red Car	Yes
B	Car with Reg. No. AB11_XYZ hit a Red BMW on Oxford Street	Yes
C	Green Car with Reg AB11_XYZ hit Red BMW on Oxford Street	No
D	Blue Car with Reg. No. AB11_XYZ hit a Red Car on Oxford Street	Yes
E	Blue Car with Reg. No. AB11_XYZ hit a Red Car in Oxford	Yes
F	A Silver Car with Reg. No. PQ11_RST hit another car on Cambridge Street	Yes

*Continued on next page*

Table 9.3 – Continued from previous page

ID	Proposition	Satisfiable
G	James's car was involved in an accident on Cambridge St at 10:30 AM	Yes
H	James had an accident with his car in Cambridge at 10:30 AM	Yes
I	James has died in a road accident this morning at 10:30 AM	Yes
J	A Red Car has collided with a Silver car on Cambridge Street	Yes

Table 9.4: Summary of Information Presented in Table 9.1, 9.2 and 9.3

ID	Proposition	Satisfiable	Scores		
			Location	Freshness	Reputation
A	Blue Nissan hit another Red Car	Yes	4	4	5
B	Car with Reg. No. AB11_XYZ hit a Red BMW on Oxford Street	Yes	3	1	8
C	Green Car with Reg. No. AB11_XYZ hit Red BMW on Oxford Street	No	5	7	-2
D	Blue Car with Reg. No. AB11_XYZ hit a Red Car on Oxford Street	Yes	5	3	5
E	Blue Car with Reg. No. AB11_XYZ hit a Red Car in Oxford	Yes	7	5	3
F	A Silver Car with Reg. No. PQ11_RST hit another car on Cambridge Street	Yes	3	3	2
G	James's car was involved in an accident on Cambridge St at 10:30 AM	Yes	5	8	4

*Continued on next page*

Table 9.4 – Continued from previous page

ID	Proposition	Satisfiable	Scores		
			Location	Freshness	Reputation
H	James had an accident with his car in Cambridge at 10:30 AM	Yes	4	5	1
I	James has died in a road accident this morning at 10:30 AM	Yes	6	9	5
J	A Red Car has collided with a Silver car on Cambridge Street	Yes	4	7	3

### Construction of World Views

After checking the satisfiability of each individual message and the result (i.e. whether they are consistent with the known facts) is stored in the database, a world view is constructed by combining the available messages together. Different combinations of the available messages are (automatically) fed into the DSS (Unit 5 on the System Framework as shown in Figure 7.1) to check if any of the combinations are found satisfiable. The combination of messages or message clusters that form a world view is stored in the database. After completing the process of generating world views, the DSS produces the output as shown in Figure 9.3. (The complete process of constructing world views is described elaborately in Section 7.4.)

### Scoring the World Views

After having the world views generated, we need to score the world views before visualising them or presenting them to others. However, the DSS performs this scoring automatically by using the data shown in Table 9.4. The scoring procedure of the world views (i.e. how the raw scores should be combined) is described in Section 5.1.5.2 and summarised in Table 5.6. Thus, by performing the following calculations on the scores shown in Table 9.4:

1. Root Mean Square of Location scores
2. Arithmetic Mean of Freshness and Reputation scores

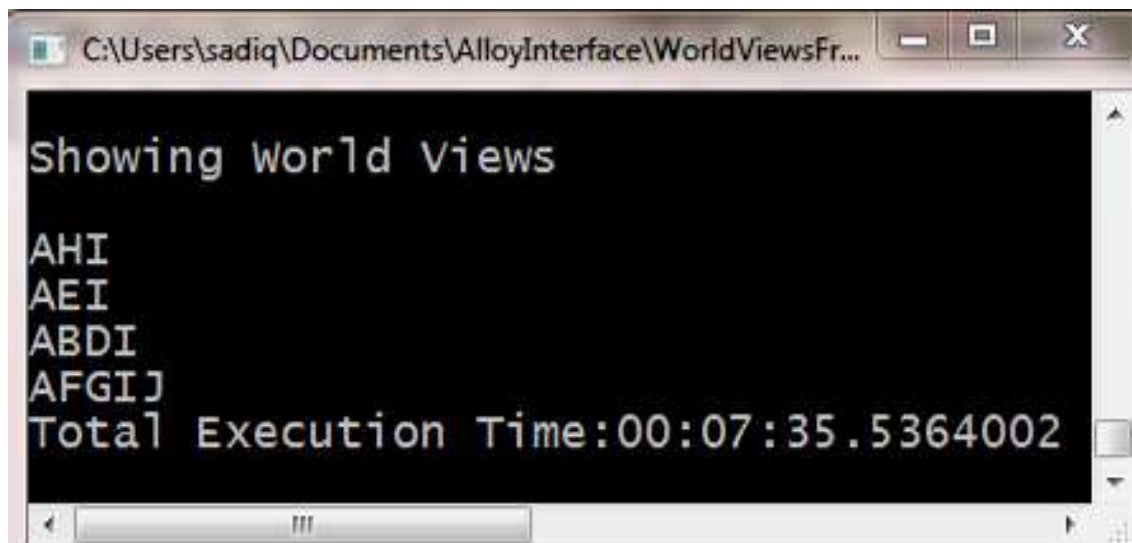


Figure 9.3: World Views Generated from the 10 Messages Listed in Table 9.1

3. Arithmetic Mean of the aggregate scores or averages found in the previous steps

we get the aggregate scores for each of the trust factors and the overall score of a world view. Table 9.5 shows the working out of the overall score of the world view **AFGIJ**.

Table 9.5: Calculation of the Overall Score of the World View AFGIJ

Msg ID	Location	Freshness	Reputation
A	4	4	5
F	3	3	2
G	5	8	4
I	6	9	5
J	4	7	3
<b>RMS = 4.52</b>		<b>Mean = 6.2</b>	<b>Mean = 3.8</b>
<b>Overall Score of the World View AFGIJ =</b>			<b>4.84</b>

Thereafter, the DSS produces the graphical representation of the world views, as shown in Figure 9.4. Scores of each of the four world views (produced by the DSS from the above ten messages) and the messages included in each of them are shown in Table 9.6.

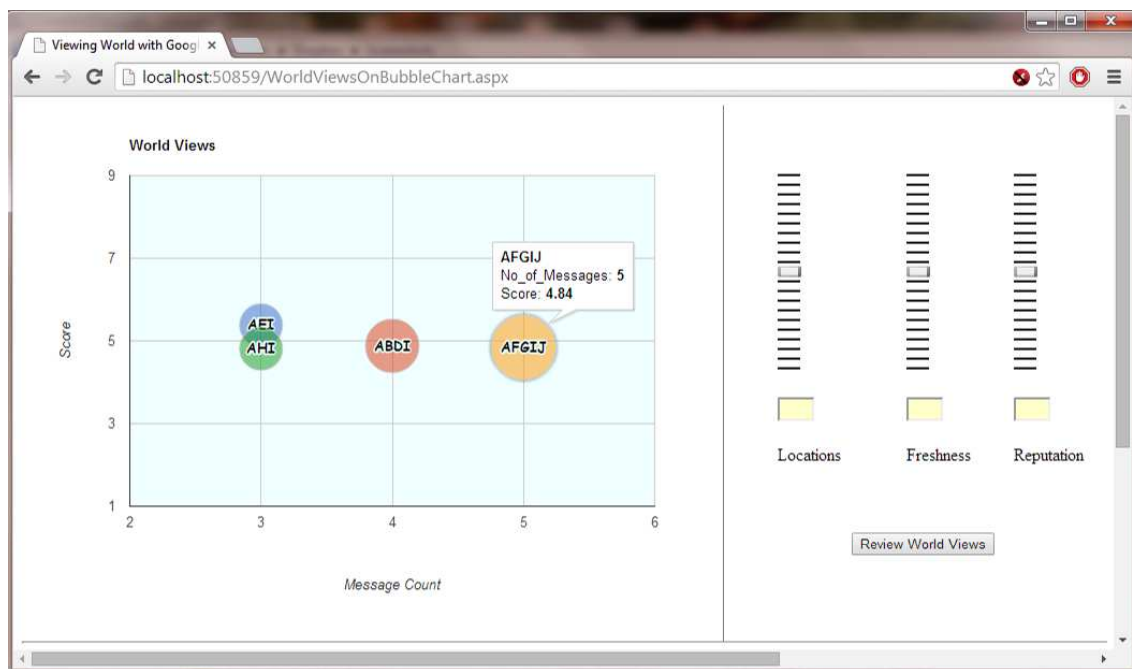


Figure 9.4: World Views Generated by the Decision Support System from the Messages Listed in Table 9.1

## 9.2 Experiments

Experiments have been carried out to answer the questions and demonstrate the expected properties of the world views that are listed in the beginning of this chapter. The experiment results have been presented and discussed in the same order of the questions/properties outlined above.

### 9.2.1 Is the Consistency Analysis Performed by DSS Correct?

The result produced by the DSS shows that each of the messages except the third one (Message-C i.e. the message with ID C) in Table 9.4 is satisfiable. The assertion that the Message-C is not satisfiable is correct because Message-C claims that the car with registration number AB11\_XYZ is green whereas, according to the known facts, the car with registration number AB11\_XYZ is blue, not green. The other messages do not contradict with any of the facts, for example, Message-D and Message-E claimed that the above mentioned car is blue which is true according to the known facts. Hence, the rest of the messages are satisfiable.

Table 9.6: World Views Generated from the 10 Messages Shown in Table 9.1

World View	Messages in World Views	Score
A F G I J	A Blue Nissan hit another Red Car F A Silver Car with Reg. No. PQ11_RST hit another car on Cambridge Street G James's car was involved in an accident on Cambridge St at 10:30 AM I James has died in a road accident this morning at 10:30 AM J A Red Car has collided with a Silver car on Cambridge Street	4.84
A B D I	A Blue Nissan hit another Red Car B Car with Reg. No. AB11_XYZ hit a Red BMW on Oxford Street D Blue Car with Reg. No. AB11_XYZ hit a Red Car on Oxford Street I James has died in a road accident this morning at 10:30 AM	4.88
A E I	A Blue Nissan hit another Red Car E Blue Car with Reg. No. AB11_XYZ hit a Red Car in Oxford I James has died in a road accident this morning at 10:30 AM	5.38
A H I	A Blue Nissan hit another Red Car H James had an accident with his car in Cambridge at 10:30 AM I James has died in a road accident this morning at 10:30 AM	4.81

### 9.2.2 Logical Soundness of the World Views

Two of the world views listed in Table 9.6 are 'ABDI' and 'AEI'. Question may be asked, why the system constructed two smaller world views ('ABDI' and 'AEI') instead of constructing a larger world view 'ABDEI'. It appears that message 'E' is in conflict with message 'B' and message 'D', as the other two messages, 'A', and 'I', are common in both world views. Now, the question is why the message 'E' is in conflict with those two messages (message 'B' and message 'D'). This is what each of the messages say:



**Message - B:** Car with Reg. No. AB11\_XYZ hit a Red BMW on Oxford Street

**Message - D:** Blue Car with Reg. No. AB11\_XYZ hit a Red Car on Oxford Street

**Message - E:** Blue Car with Reg. No. AB11\_XYZ hit a Red Car in Oxford

Fact- 2 states that an object cannot be at two different locations at the same time, and according to Fact- 20 and Fact- 21, Oxford Street and Oxford are two different locations. Accordingly, Message-B and Message-D are in conflict with Message-E as the first two messages claim that the Car with registration number AB11\_XYZ was involved in an accident at Oxford Street, while Message-E says that the same car was involved in an accident in Oxford.

Hence, the experiment result demonstrates that every new message will have either of the following two effects on world views:

- It will be included in one or more existing world views and (thus,) it will make those world views richer by being supportive of them
- It will increase the number of world views by forming new world views together with other messages or on its own.

The experiment result also demonstrates another property of world views that the size and number of existing world views never decrease with the increase of input messages.

**Is the set of world views generated by the DSS a universal set for the given messages?**

The set of world views generated by the DSS is the universal set (of world views) for the given messages. Manual processing of the input messages shown in Table 9.4 produced the same set of world views as the decision support system generated. The world views could not be anything different and their number could not be more or less. Hence, the set of world views generated by the DSS is the universal set (of world views) for the given messages i.e. the DSS generates all possible world views.

### 9.2.3 Impact of a new message on existing world views

The decision support system has generated four world views from ten messages. It should be noted that if the data set did not include Message-H and Message-J, then



Figure 9.5: Impact of Decision Making Policy on World Views (a)

there would be three world views, such as ‘ABDI’, ‘AEI’ and ‘AFGI’. Message-H has formed a new world view ‘AHI’ as it does not satisfy with any of the existing world views. On the contrary, Message-J is satisfiable with existing world view ‘AFGI’ and therefore, it has formed a larger world view ‘AFGIJ’ by assimilating with ‘AFGI’.

#### 9.2.4 Decision Making Policy - Sensitivity Analysis

Considering some of the human factors, the Decision Support System has incorporated the Decision Making Policy discussed in Section 5.1.6. If any of the information provenance and quality factors appears to be more (or less) important to the decision maker, then s/he is likely to accentuate (or attenuate) that factor. Figure 9.5 Figure 9.6 and Figure 9.7 show how the overall score of world views change when any of the factors are given more (or less) importance than others. However, the Decision Making Policy may not always change the order of their reliability (as shown in Figure 9.7).

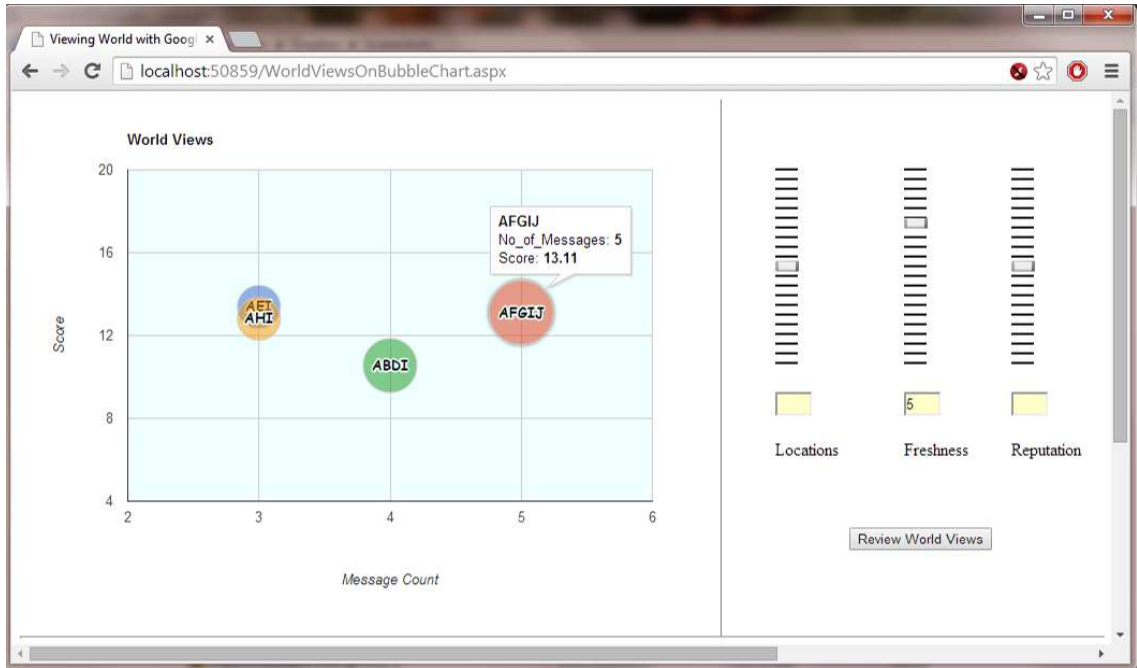


Figure 9.6: Impact of Decision Making Policy on World Views (b)



Figure 9.7: Impact of Decision Making Policy on World Views (c)

### 9.2.5 Constructing World Views with a Larger Data Set

In subsection 9.1.5, world views were constructed from ten messages. In this experiment, the decision support system is given twenty messages i.e. all of the messages listed in Table 9.4 and Table 9.7 to construct world views using the same set of facts described in subsection 9.1.5.

Table 9.7: Test Data Set-2: The Messages Used to Construct World Views and their Satisfiability Evaluated by the DSS

ID	Proposition	Satisfiable	Scores		
			Location	Freshness	Reputation
K	Traffic accident on Cambridge Street with no casualty	Yes	5	5	6
L	James Sent a Message at 5:00 PM	Yes	6	6	4
M	Traffic collision at Oxford Street, both drives are spot dead	Yes	1	5	1
N	Three people died after traffic collision at Oxford Street	Yes	2	4	5
O	Traffic collision at Oxford Street, 3 people seriously injured	Yes	4	2	3
P	Three people died in a traffic accident in Westminster	Yes	5	5	5
Q	Traffic accident in Westminster 1 person died	Yes	3	3	1
R	Car Drove into a bus stop shelter at Oxford Street. 3 people seriously injured	Yes	2	4	2

*Continued on next page*

Table 9.7 – Continued from previous page

ID	Proposition	Satisfiable	Scores		
			Location	Freshness	Reputation
S	James was somewhere in Cambridge Street at 10 30am	Yes	4	3	4
T	James car hit another car at Cambridge Street at 10 30am	Yes	3	1	0

The world views constructed by the decision support system in this experiment are listed in Table 9.8 along with the total score and the number of messages included in each world view. A more detailed list of these world views, which shows the messages included in each world view separately, is given in Appendix B.2. Graphical representation of these world views is also shown in Figure 9.8. Notice that on Figure 9.8, the scores of the world views have been multiplied with ten for increasing their differences. Nevertheless, some of the world views are overlapping each other for having the same number of messages and very similar scores. In Figure 9.9, although most world views are spaced apart and nicely visible, one of them (**ABDLM**) is completely invisible for being stacked under the world view **ABDLQ**.

Table 9.8: World Views Generated from the Test Data in Table 9.4 and Table 9.7

World View	No. of Messages in World View	Score
AFGIJST	7	4.18
AFGJKST	7	3.98
ABDLNP	6	4.51
ABDLOR	6	4.01
ABDIQ	5	4.39
ABDLM	5	4.19
ABDLQ	5	4.12
AFJKL	5	4.51
AEI	3	5.38
AEL	3	4.93
AHI	3	4.81

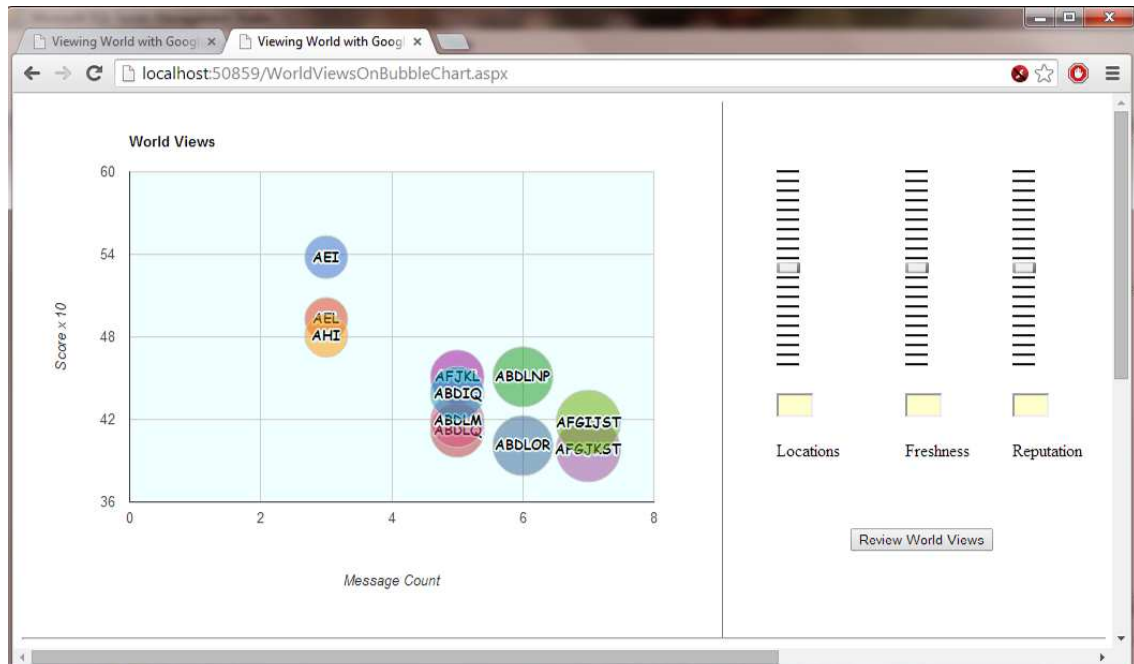


Figure 9.8: World Views Generated from 20 Messages



Figure 9.9: Change in Score has Reduced Overlapping of the World Views

### 9.2.6 World Views Adhere to the Logical Rules – if a premise is satisfiable, then its consequent is also satisfiable

One of the facts (Fact -15) used to construct world views state that James does not allow anyone else to drive his car. This means, if James's car is on the street (i.e. being driven), then James is there wherever the car is (assuming that cars do not get stolen or towed away). Hence, Message-S (James was somewhere in Cambridge Street at 10 30am) is a logical consequence of Message-T (James car hit another car at Cambridge Street at 10 30am). Accordingly, any world view that includes Message-T must include Message-S too and there may be some world views that contains Message-S but not Message-T.

The world views generated by the decision support system (listed in Table 9.8) demonstrate that they adhere to the above mentioned logical inference rule as the two world views 'AFGLJST' and 'AFGJKST' contain both Message-T and Message-S and there is no other world view that contains Message-T. However, there are examples of world views ('AIS' and 'AKS' in Table B.3 in Appendix B) that contain Message-S but not the Message-T.

## 9.3 Processing Capacity of Alloy Analyzer and its Suitability in Real Life Cases

The methodology of constructing world views from available messages has been discussed in detail in Chapter 7. World views may be generated in two modes: Batch Mode and Serial or Incremental Mode. In batch mode, all individually satisfiable messages are taken together from the available messages and their collective satisfiability is checked. If all of these messages are found satisfiable together, then these messages form a world view. If all of these messages do satisfy together, then smaller clusters are checked to see if they form world views. This process of pruning and checking the satisfiability of the message-clusters continues until a world view is found. In Serial or Incremental Mode, a single message is taken first. Then another message is combined with the first message. If these two messages are found satisfiable together, then a third message is combined with them to form a larger world view. Thus, the size of the cluster of messages increases over time as the process continues. Which mode of operation will be quicker to find the world views depends on how consistent the messages are with each other. The batch mode will

definitely be quicker if all the messages are consistent with each other. The serial mode is likely to be quicker if most of the messages are inconsistent with each other. However, whichever mode of operation is chosen, the set of world views will always be the same.

I have processed the first ten messages in both incremental (or serial) and batch mode. The computer used for carrying out all of the above experiments is a very small laptop, which has the following specification:

**Processor:** Intel Atom CPU @ 1.66GHz

**Memory:** 2.00 GB

**Type of Storage Device:** Solid-state drive

**Operating System:** Windows 7 Enterprise (32 bit)

In batch mode, the decision support system checked the satisfiability of 464 message-clusters before finding all (four) possible world views. The whole process took approximately 7 minutes and 35.54 seconds to complete (see Figure 9.10). On the other hand, the application checked the satisfiability of 67 message-clusters before generating all possible world views in serial mode and it took approximately 1 minute and 29.64 seconds (see Figure 9.11) to complete the process.

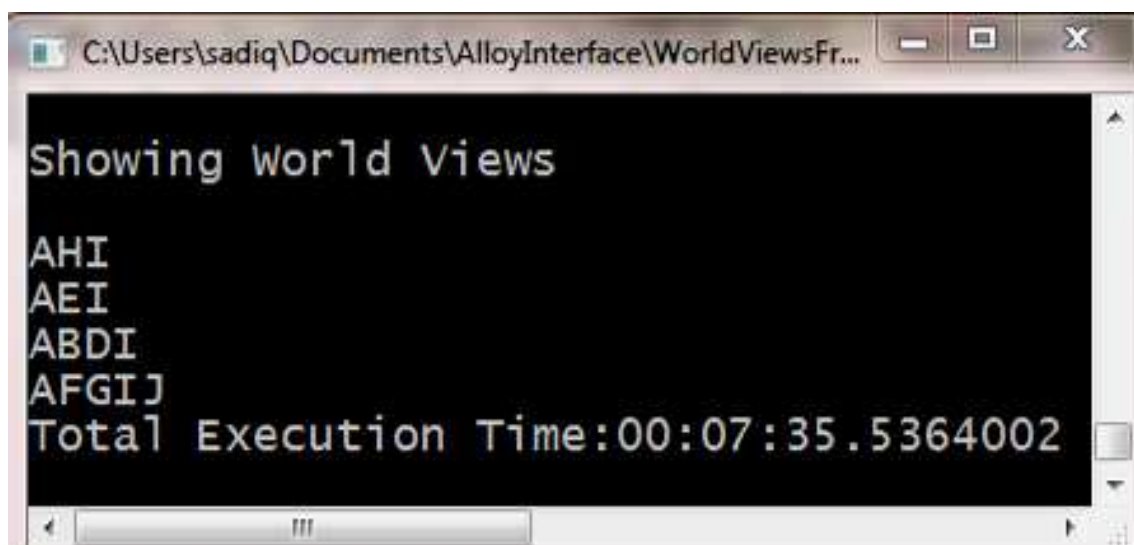


Figure 9.10: Time Taken to Generate World Views from 10 Messages in Batch Mode



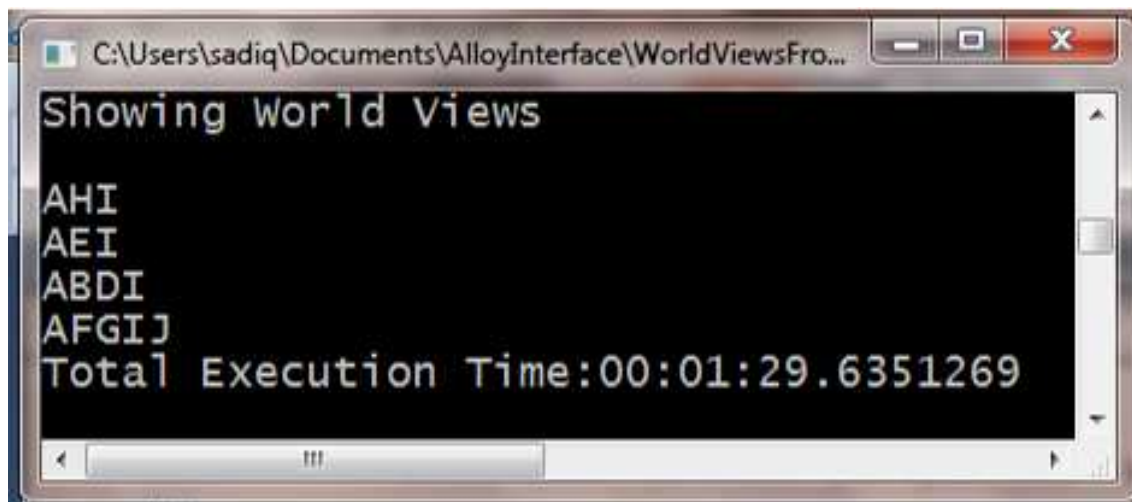


Figure 9.11: Time Taken to Generate World Views from 10 Messages in Serial Mode

The application took 8 minutes and 24.07 seconds (approx.) for generating the world views shown in Table 9.8 from twenty messages in incremental or serial mode. Satisfiability of 485 message-clusters were checked before finding the final fourteen world views. It should be noted that the time taken to process twenty messages in serial mode is less than a minute more than the time taken to process ten messages in batch mode. The number of message-clusters checked for satisfiability in both methods is not very different (480 and 485) either, even though the number of input messages is exactly half than in the other (10 and 20). The experiment result shows that for the given data set, serial or incremental mode of operation is more efficient than batch mode.

To understand the processing capacity or scalability of Alloy Analyzer based Decision Support System, I have carried out another experiment with twenty five messages (twenty messages listed in Table 9.4 and Table 9.7 plus five new messages) shown in Table B.1. The application checked the satisfiability of 760 message-clusters before generating twenty world views. The whole process took approximately 12 minutes and 13.63 seconds (see Figure 9.12). I tried to process the first twenty messages in batch mode on the same computer. However, the application, which took about 8 minutes to finish processing the same twenty messages and about 12 minutes to process twenty-five messages in serial mode, stopped unexpectedly after continuously running for more than 12 hours. In this 12 hours time, the application checked the satisfiability of 33,866 message-clusters.

Later, I have run the application on a more powerful computer to process the

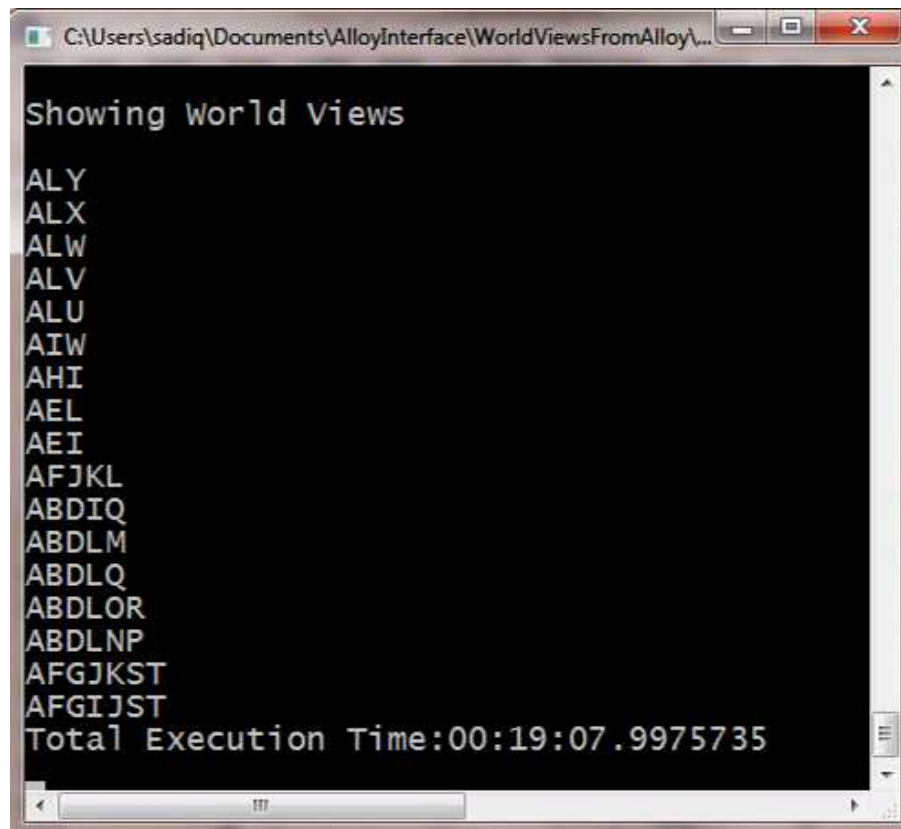


Figure 9.12: Time Taken to Generate World Views from 25 Messages in Serial Mode

twenty-five messages. The computer used to process twenty-five messages has the following specifications:

**Processor:** Intel Core i7 (16 core) Processor @ 2.66GHz over clocked to 4.7GHz

**Memory:** 8.00 GB

**Type of Storage Device:** Solid-state drive

**Operating System:** Windows 7 Enterprise (32 bit)

Nevertheless, the application (running on this powerful machine ) could not find a single satisfiable cluster before it was stopped after running for about 20 hours (from 22/06/2013 18:50:51 to 23/06/2013 14:44:25). In this 20 hours time, the application checked the satisfiability of 185,510 message-clusters.

It should be noted that

- The application is using breadth-first search to find a world view.
- The largest world views (e.g. ‘ABDLNP’, ‘ABDLOR’, etc.) that can be generated from these 25 messages has length 6 i.e. they contain six messages (see Figure 9.12).
- The application started its process with the message-cluster ‘ABCDEFGHIJKLMNOPQRSTUVWXYZ’ (message ‘C’ is missing from the cluster because it is intrinsically unsatisfiable) and the last message-cluster (‘BDEGHJKMNQRSTUVWXYZ’) processed by the DSS before it was stopped after running for twenty hours is of length 18. This means, the application took about twenty hours to prune the message-cluster and reduce its length from 25 to 18.

Hence, we can imagine how long the application would take to further prune the message-cluster and bring its length down from 18 to 6 before finding any of the largest world views. It may be argued that the application would take less time if a depth-first search was used, in this case (i.e. for processing this or similar data-set). However, the depth-first search is likely to be unsuitable for a different data-set because the suitability of depth-first search and breadth-first search depends on the data-set. However, the time complexity of depth-first search and breadth-first search is exactly the same in the worst case scenario, as the entire space needs to be searched. Whether other search algorithms can be used in finding the world views more quickly may be examined in the future.

What does this signify?

Even if the application runs in serial mode to construct world views from a reasonably larger set of messages, it is very likely that the application will generate one or more world views of size 24 (i.e. containing twenty-four messages), for example. If an additional message needs to be processed and combined with the existing world views after having a world view containing twenty-four messages, then this is essentially the same case as processing twenty-five messages in batch mode. Although the time that may be taken to process these twenty-five messages depends on how consistent the new message is with the messages included in the world view, it is not unlikely that the new message is not consistent with any of these messages. In such worst case scenario involving twenty-five messages, the application needs to check the satisfiability of  $\sum_{r=1}^n n_{c_r}$  ( $= 33,554,431$ ) distinct message-clusters. Whereas, the

application has taken approximately 20 hours to check the satisfiability of 185,510 message-clusters only. Hence, we can imagine how long the application will take to check the entire search space (i.e. check the satisfiability of 33,554,431 message-clusters). Even if the new (additional) message does not create the worst case scenario, the application may take much longer than twenty hours to create world views from twenty-five messages as the last experiment result indicates.

This unreasonably long processing time points to the problem of state explosion and highlights one of the scalability issues of the decision support system. In real-life cases, the application is expected to handle thousands of messages, if not millions. It is very unlikely that the current Alloy Analyzer based DSS running on a single machine will be able to perform to a satisfactory level. However, this particular scalability problem may be mitigated by using parallel processing, distributed and cloud computing.

## 9.4 Summary

This chapter demonstrates how a user can use the Decision Support System, step by step, using some contrived data. As Figure 7.1 shows that the Decision Support System has multiple components and the output of one component is the input to another component, the step by step demonstration also describes what the input, output and their data formats are, at each stage.

This chapter also demonstrates various technical aspects of the Decision Support System and some interesting phenomena of the world views. The experimental results show that automated reasoning and consistency checking of messages using Alloy can help us constructing world views from untrusted information. The results from the experiments also show that the world view containing the maximum number of messages may not necessarily form the most reliable world view (see Table 9.6). This may foil the bid to fool people by corroborating certain messages by a syndicate. The technical aspects of the DSS that the experiment result demonstrates include:

1. The consistency evaluation of each individual message performed by the DSS is sound.
2. The DSS generates the complete set of unique and largest possible world views from the available messages.

3. The world views generated by the DSS are logically sound and follows logical inference rules such as if a premise is satisfiable, then its consequent is satisfiable too.
4. Every new message somehow makes a change to the existing world views.
5. The Decision Making Policy affect the ranking of the world views.

The experiment results also showed (in Section 9.3) that the construction of world views was much faster in serial mode than in batch mode for the same data set. For constructing world views from ten messages, the DSS took more than 7 minutes in batch mode whereas, it took less than 2 minutes (approximately) to generate the same world views in serial mode. However, it is not guaranteed that the construction of world views in serial mode will always be quicker. As with the case of depth-first search and breadth-first search for satisfiable clusters or world views, whether the serial mode or batch mode of operation will be faster depends on the nature of the data set.

The performance and scalability issues of the Alloy Analyzer based DSS has also been highlighted by the experiment results. However, this performance and scalability issues may be mitigated using parallel processing, distributed and cloud computing.

The experiment results also show that some of the world views are very similar with little differences. For example, world views **AFGIJST** and **AFGJKST** (shown in Table 9.8) have six messages (**A, F, G, J, S, and T**) common in them. It may be argued that if a fuzzy world view was constructed including those similar world views, then it might have given a better view of a situation. A detailed discussion on consistent versus fuzzy world views has been given in the following chapter.

## Conclusion and Future Work

In every 24 hours, 294 billion emails are sent, 2 million blog posts are written, 172 million different people visit Facebook, 40 million people visit Twitter, 532 million statuses are updated on Facebook, 250 million photos are updated on Facebook, and 864,000 hours of videos are uploaded to YouTube alone [126]. This gives a glimpse of how much information is being uploaded to the Internet each day. However, the amount of information added to the Internet by the people from a particular region increases many fold in case of a large-scale disaster and other kind of events or incidents that affect their lives. It is impossible to process all this data manually. In addition, the information available on the Internet contain both good and bad information (misinformation and disinformation), as shown in Chapter 3. Therefore, it is necessary to curate the information before using it. When it comes to dealing with a crisis or emergency situation, the importance of distinguishing between reliable and unreliable information is even higher. Hence, the emergency responders and decision makers need an automated decision support system that can help them using the ‘big data’.

The case studies carried out using the data automatically harvested from the micro-blogging service Twitter, give an insight about the quality and utility of such information. The case studies reveal that although such information contain a large amount of noise and conflicting reports, the content of the tweets converge over a period of time and thereby remove all uncertainties. Therefore, the aim of this research is to investigate how the varying provenance of the data can be tracked and exploited to prioritise the information presented to a busy incident controller and to synthesise a model or models of the situation that the evidence pertains to.

The novelty of this research stems from its innovative approach to develop a system to allow the decision makers to augment official information with the in-

formation contributed by the wider public (either explicitly submitted to them or harvested from social networks such as Facebook and Twitter), and to be able to handle inconsistencies and uncertainty arising from the unreliability of such sources in a flexible way. The system will also provide decision support and will help to improve situational awareness, especially in a crisis and emergency situation.

The system and techniques, developed from this research, involves

- Data harvesting from open-sources like social networks (e.g. Twitter, Facebook, etc.)
- Prioritising the information based on its provenance and other information quality factors that influence trust
- Knowledge based reasoning for consistency analysis and conflict resolution

with an ultimate purpose of constructing one or more world views that the evidence pertains to.

Data harvesting can be performed targeting a specific location, time frame and individuals. Data harvesting can also be generalised simply using keyword search without specifying a target (individual, time or location). Since the harvested data, in general, is more likely to be written in a plain natural language and the tweets (Twitter Data), in particular, are written in plain text as shown in Appendix A.1, the harvested data is transformed into an intermediate structured format, namely the Tactical Situation Object (TSO), before further processing.

Provenance and other information quality and human trust factors influence people's trust in information. Provenance (also referred to as lineage and pedigree) of information refers to the source of information such as who gave (or produced) the information, the derivation history of information, what data was used to generate it, and also finding the trail of how the information has passed from one source to the other and how it has been changed. Provenance of information has direct and indirect links with security properties, especially with information authenticity, integrity and access control [80]. Thus, provenance information helps to assess the quality of information (correctness, authenticity, integrity, etc.) and thereby, helps to determine the level of trust that can be attributed to it [56, 140]. A detailed discussion on different factors, including provenance and quality of information, that influence trust in information has appeared in Chapter 4.

Another novel aspect of this research is the use of Alloy to perform an automated logical consistency analysis in order to resolve contradiction between messages and

to construct world views. Generally, logical reasoning and specifically, Alloy has been widely used in various applications. However, the use of Alloy and Alloy Analyzer (sic) for checking the consistency of messages, especially in the context of crisis and emergency management is a novel approach. A detailed discussion on this novelty has appeared in the chapter ‘Managing Crisis Information - State of the Art’ (Chapter 2).

The decision support system has been implemented incorporating all novel aspects of the research. Although the system has some limitations that could not be overcome with the current state of technology such as Natural Language Processing, which is beyond the scope of this research, the world views generated from the test-data are found logically sound and follows the logical inference rules that are applicable. The output of the DSS also demonstrates the proof of concept used in the decision support system. The world views generated by the DSS present only that information which is consistent with our existing knowledge and is related to the concerned location. The output of the DSS also demonstrated that the reliability of a world view may appear to be low because of the provenance and other trust metric, even if a world view contains more messages (i.e. more corroboration) than others. This will reduce the effect of possible collusion i.e. this may foil the bid to manipulate the output of the DSS by corroborating certain messages by a syndicate.

## 10.1 Consistent Vs Fuzzy World Views: A Critical Analysis

The objective of constructing world views is to produce the best possible picture(s) of a situation from the available information and to present them to the user in a tractable manner. However, in a medium or large scale crisis and emergency situation, there is likely to be a large amount of messages that are (largely in agreement but) in partial disagreement with each other on smaller issues. Hence, an attempt to construct totally consistent world views may not produce an expected result. In real life cases when the number of input messages is expected to be enormous, an attempt to construct totally consistent world views may lead to a very large number of smaller and similar world views, each containing relatively a small number of messages. These smaller world views are less likely to produce a comprehensive picture of a situation and handling the information may still be intractable due to the large number of world views. To produce a comprehensive picture of a situation



and to keep the number of world views tractable, it may be necessary to construct slightly fuzzy views of a situation instead of fully consistent views.

Some of the world views generated by the decision support system (in Chapter 9) are very similar. For example, world views **AFGIJST** and **AFGJKST** (shown in Table 9.8) have six messages (**A, F, G, J, S, and T**) common in them. Instead of making two smaller world views, it would be possible to form a larger world view 'AFGIJKST', if the Message-I and Message-K did not contradict. Here is what these two messages claim:

**Message - I:** James has died in a road accident this morning at 10:30 AM

**Message - K:** Traffic accident on Cambridge Street with no casualty

Although, both of the messages agree that there is a traffic accident but they differ in the number of casualty. Because of the same reason that they differ in both the number and description of casualties, messages 'M', 'N', 'O', 'P', 'Q', and 'R' could not fit in the same world view. It can be argued that all these messages (M, N, O, P, Q, and R) are reporting about the same incident, because all of them are referring to a traffic incident at the same location. If these messages really refer to the same incident, then arguably, these messages should be included in the same world view even though they disagree with each other as to the type and number of casualty. In that case, a fuzzy world view (as opposed to a fully consistent world view) may arguably provide a better situation awareness.

Similarly, world views 'ABDI' and 'AEI' (generated from ten messages as shown in Table 9.6) are different world views only because one of the messages (Message-E) states that a car was involved in an accident in Oxford while the others claim that the same vehicle was involved in an accident in Oxford Street. The sender of the message 'E' might have wanted to write Oxford Street as the accident location but possibly, because of rush and/or nervousness, s/he only wrote Oxford, instead of Oxford Street. Hence, presenting fuzzy world views rather than strictly consistent world views may be more useful in some cases. A possible way of presenting a fuzzy world view (which includes three consistent world views ABDI, AEI and AHI) is shown in Figure 10.1 with a dotted circle around it.

Despite having some limitations as mentioned earlier, the approach used to create the picture of a situation, described in Section 7.1, essentially presents both fuzzy and consistent view of a situation. A lower level of the tree, shown in Figure 7.3 and Figure 10.2, gives a more consistent view of a situation than its upper levels.

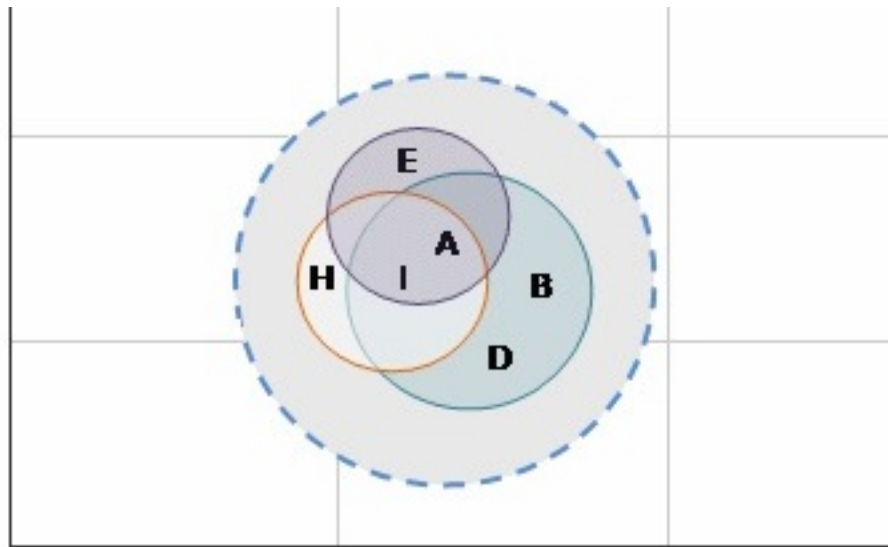


Figure 10.1: A Fuzzy World View

Conversely, an upper level on the tree gives a larger but more fuzzy view of a situation than its lower levels. For example, in Figure 10.2, all of the 29 messages unanimously report about a transport incident, whereas 25 of them are reporting that a traffic incident has taken place in Oxford Street and 18 out of these 25 messages say that a traffic incident involving two vehicles has taken place in Oxford Street.

An alternative way of constructing fuzzy world views is to use the concept of fuzzy sets [166]. In classical sets, elements either belong to a set or do not belong to a set; there is no other option in between. However, in Fuzzy Logic, fuzzy sets are defined using a membership function, which makes it possible for an element to belong to a set partially. So, a fuzzy set is a set that is defined by a membership function and the membership function assigns a level/grade of membership to each element in the set under consideration (universe of discourse). Membership grades range from 0 (zero) to 1. For example, according to the classical set theory, if a message is not one hundred percent consistent with (the messages in) a world view, the message will be called inconsistent and cannot be included in that world view. However, using the fuzzy set and its membership function, we can say that the membership grade of the message is 0.8 (say) with respect to that world view. That is, the message is almost consistent with the world view. The same message can have different membership grades for different world views, depending on how much consistent/inconsistent it is with those world views.

Suppose a world view says, there is a traffic accident involving two vehicles

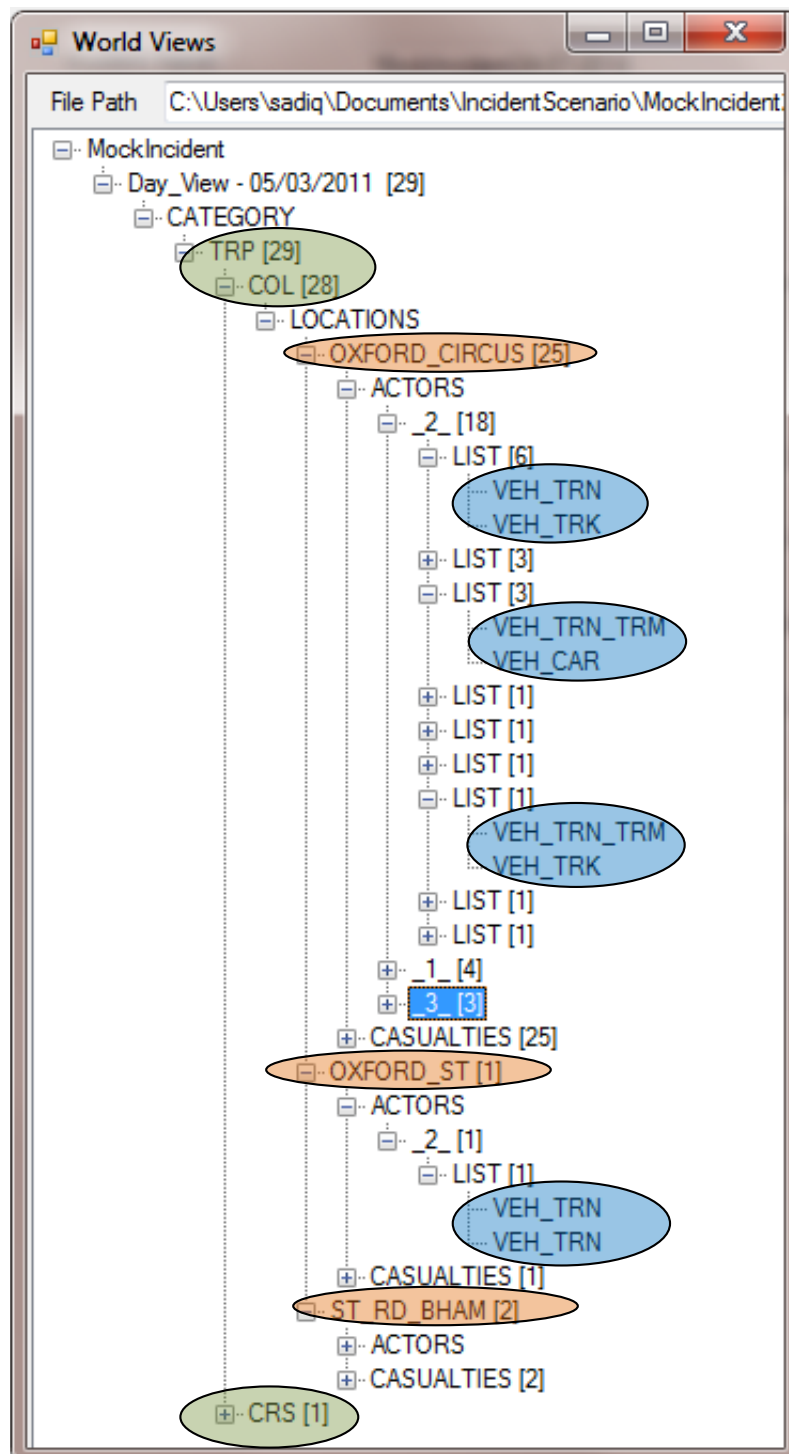


Figure 10.2: World Views: Consistent and Fuzzy, Two-in-One View

on Oxford Street with three casualties. Recall that some of the main components of TSO that describes an event are Event, Date and Time, Location, Actor and Casualty. Suppose, the specification of a membership function defines the grade of membership of a message to a world view as follows:

- Event: 0.25
- Date and Time: 0.25
- Location: 0.25
- Actor: 0.15
- Casualty: 0.10

That is, if a message agrees only with the description of the event found in the world view (i.e. what the event is), then it will receive a membership grade 0.25. Hence, according to this definition, a message (not included in the world view mentioned above), which says, '*there is a traffic accident involving two cars and a bike on Oxford Street with one casualty*', can be a partial member of the world view with a membership grade 0.75 (as the message does not agree with the world view on the number of actors and casualty;  $0.25+0.25+0.25+0+0 = 0.75$ ). The grading score for each of the TSO components may be subdivided into smaller grades corresponding to their subcomponents. For example, the membership grade for Date and Time may be subdivided into 0.15 and 0.10 for date and time respectively. Thus, after having generated the fully consistent world views from a set of messages, a decision maker can select a particular world view to construct a fuzzy (world) view by including in it the messages that have a membership grade 0.7 or above, for example. A fuzzy world view may be created using the specified membership grade for each of the fully consistent world views generated, or only for a single world view chosen by the decision maker.

## 10.2 Limitations and Future Work

The case study on Haiti earthquake detailed in Chapter 3 reveals that people implicitly expose their location through their tweets (see Table 3.4). However, it will be necessary to analyse the message context (text analysis) and use a strong algorithm to extract the location information from explicit or implicit mentioning of a place,

time, point of interest or general comments like “Eiffel Tower looks awesome from here”. This is certainly a topic for future research.

Although a schema has been developed for TSO, a bottleneck in the system is that messages still need to be encoded into TSO manually. Whether TSO can be automatically generated using Natural Language Processing (NLP) is also a topic for future research. Another similar problem is that the messages need to be specified manually in Alloy specification language. However, although it has not been tried yet, it should be possible to encode messages into Alloy automatically as long as the facts and rules are defined adequately for a specific context.

It has been discussed in Chapter 7 that irrelevant messages needs to be separated to reduce noise before consistency analysis is performed. Although TSO makes it easy to separate irrelevant messages i.e. clustering the relevant messages together, it may be necessary to use tools like Apache Solr, an open-source search engine, and Carrot2, an open-source search results clustering engine, in the future to cluster the relevant messages together before TSOs are created. Clustering of the relevant messages, however, will not affect the subsequent construction and use of TSOs. In fact, the clustering of the messages can also be performed after encoding them into TSOs. Whether the clustering of the messages should be performed before or after encoding them into TSOs is immaterial, except that the user may choose not to encode the irrelevant messages into TSOs (if they think it is unnecessary to process these messages), which will save time and effort.

While Table 8.1 in Chapter 8 has outlined the interfaces between different components (i.e. what the inputs and outputs of each of the components are and their data format) of the DSS, Chapter 9 has demonstrated the application of the whole framework and how the user can use the system, step by step. Experiments have also been carried out with the decision support system using some contrived data (in Chapter 9). Although the experiment result demonstrated the proof of concept, the DSS needs to be tried in a range of real-life cases, in order to understand the utility of the system. However, the experiment results indicate that the Alloy Analyzer has a limitation. It may not be feasible to use Alloy Analyzer for consistency analysis of reasonable amount of messages unless other advanced techniques (such as parallel processing, distributed and cloud computing, etc.) are used. The MIT-developed Alloy Analyzer may also need further improvement in order to address the scalability issue.

Categorising messages into different groups according to the location and type of incident will reduce the possibility of getting unrelated messages in the same cluster

or world view. However, defining the area or scope of a specific location is a real challenge. Because, for example, if there is an incident in Oxford Circus, London, some people may simply refer to the location as ‘Oxford Street’, ‘Westminster’, ‘West End’ or even ‘London’. If the scope of the location is restricted to the area covered by Oxford Circus or Oxford Street, then a large volume of information is likely to be ignored as they will appear to be associated with a different place i.e. out of scope. On the contrary, if London is accepted as the location, then we will receive too much noise and the world views are likely to be extremely ambiguous.

Another challenge is to determine whether the reliability/utility score of a world view should be diminished or not, when that world view needs a larger scope to satisfy. If we really decide to diminish the score, then to what extent shall we diminish the reliability of the world view when it needs a larger scope to satisfy?

My research and development have been focused on a practical and experimental demonstration of the concept and potential for such reasoning systems. However, producing a proof of the algorithmic performance of the system may be a future work. For example, current implementation of the Decision Support System uses breadth-first search to find a world view. Although I have mentioned in Chapter 9 that the suitability of depth-first search and breadth-first search depends on the data-set, whether a different (custom) search algorithm will generally be more efficient in finding the world views may be examined in the future.

The Decision Support System is designed and implemented to present the world views that are internally fully consistent i.e. the messages included in each world view are totally consistent with each other. However, it has been discussed in the previous section (Section 10.1) that slightly fuzzy world views may sometimes be more useful than strictly consistent world views to understand a situation better. Nevertheless, it is still an open question, which may be addressed in the future, as to whether it is best to present only the consistent world views or a fuzzy view would be more useful.

The future improvement of the Decision Support System should also include a visual cue to indicate that multiple world views are stacked on each other so that the user can double click on that visual cue to see all of the stacked world views. A circled number (2) in Figure 10.3 shows, as an example, that two messages are stacked on each other here.

In this work, I have made an assumption that sufficient amount of information will be available from the crowds to discover the current state of the world. In other words, I presume that the uncertainty will be caused by noise, not because

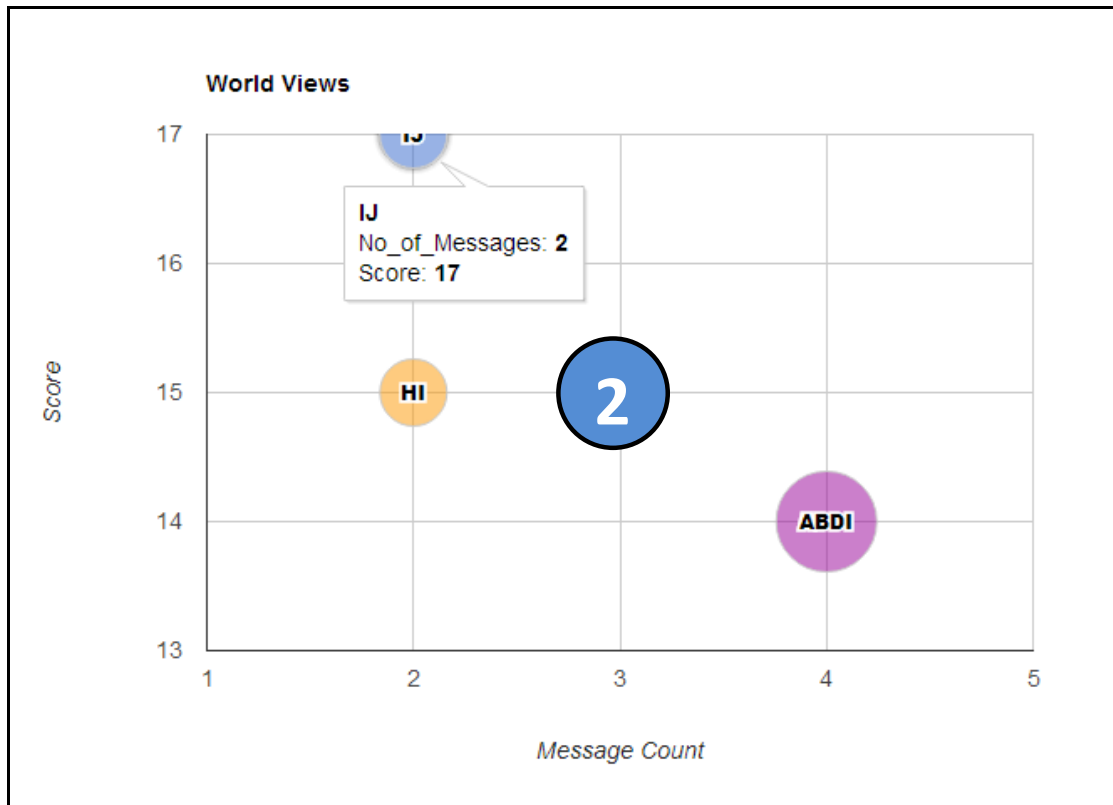


Figure 10.3: Two World Views Stacked on Each Other is Indicated by a Circled Number

of ignorance (lack of knowledge). This is why I have used predicate logic based reasoning to create multiple self-consistent world views. However, in some cases, especially when sufficient information is not available, the uncertainty will be caused by both noise and ignorance. In such cases, it may be necessary to deal with the uncertainty using an alternative and appropriate method such as symbolic methods (non-monotonic reasoning), statistical methods or Fuzzy logic methods.

The decision making policy applied by the decision maker is similar to a fine tuning method that could be supplemented with or substituted (if, it is deemed necessary in some cases) by an automated method that would apply one of the empirical and pre-set fine tuning schemes based on the type of incident. This is also a topic for the future work.

# Bibliography

- [1] Mohammad-Ali Abbasi, Shamanth Kumar, Jose Augusto Andrade Filho, and Huan Liu. Lessons learned in using social media for disaster relief - ASU crisis response game. In *Proceedings of the 5th international conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, SBP'12, pages 282–289, Berlin, Heidelberg, 2012. Springer-Verlag.
- [2] Sibel Adali, Robert Escriva, Mark K. Goldberg, Mykola Hayvanovych, Malik Magdon-Ismael, Boleslaw K. Szymanski, William A. Wallace, and Gregory Williams. Measuring behavioral trust in social networks. In *Intelligence and Security Informatics (ISI), 2010 IEEE International Conference on*, pages 150–152, May 2010.
- [3] Defense Advanced Research Projects Agency. DARPA network challenge. <http://archive.darpa.mil/networkchallenge/>. Accessed on 02/11/2012.
- [4] Defense Advanced Research Projects Agency. DARPA network challenge project report. <http://www.hSDL.org/?view&did=17522>, February 2010. Retrieved on 24/10/2012.
- [5] Mahreen Aziz Ahmad and Aamer Nadeem. Consistency checking of UML models using Description Logics: A critical review. In *6th International Conference on Emerging Technologies (ICET)*, pages 310–315, October 2010.
- [6] Myla Archer and Constance L. Heitmeyer. Human-Style Theorem Proving Using PVS. In *Proceedings of the 10th International Conference on Theorem Proving in Higher Order Logics*, TPHOLs '97, pages 33–48, London, UK, UK, 1997. Springer-Verlag.
- [7] Donovan Artz and Yolanda Gil. A survey of trust in computer science and the semantic web. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):58–71, June 2007.
- [8] Peyman Ghafari Ashtiani, Habibollah Javanmard, and Mahsa Ayoubi Moghadam. Examining The Role of Incomplete Information in Print Advertising of Travel Agencies and its Effect on Information Search, Consumer Involvement, Information Source Trustworthiness and Purchasing Intention (Case study: Iran's Tourists). In *2nd International Conference on Education and Management Technology, IPEDR*, volume 13, pages 302–306, 2011.



- [9] Kenneth Baclawski, Mieczyslaw M. Kokar, Richard Waldinger, and Paul A. Kogut. Consistency Checking of Semantic Web Ontologies. In *In First International Semantic Web Conference (ISWC 2002), Sardinia*, pages 454–459. Springer, 2002.
- [10] BBC News. As it happened: Mumbai attacks - 28 Nov. [http://news.bbc.co.uk/1/hi/world/south\\_asia/7753639.stm](http://news.bbc.co.uk/1/hi/world/south_asia/7753639.stm), 2008. Accessed on 14/02/2014.
- [11] BBC News. As it happened: Mumbai attacks - 29 Nov. [http://news.bbc.co.uk/1/hi/world/south\\_asia/7756073.stm](http://news.bbc.co.uk/1/hi/world/south_asia/7756073.stm), 2008. Accessed on 14/02/2014.
- [12] BBC News. As it happened: Mumbai attacks 27 Nov. [http://news.bbc.co.uk/1/hi/world/south\\_asia/7752003.stm](http://news.bbc.co.uk/1/hi/world/south_asia/7752003.stm), 2008. Accessed on 20/05/2010.
- [13] Christina Bellantoni and Stephen Dinan. Obama’s twitter site hacked. The Washington Times, January 2009. <http://www.washingtontimes.com/news/2009/jan/05/obamas-twitter-site-hacked/>.
- [14] Armin Biere and Robert Brummayer. Consistency Checking of All Different Constraints over Bit-Vectors within a SAT Solver. In *Formal Methods in Computer-Aided Design, 2008. FMCAD '08*, pages 1–4, November 2008.
- [15] Christian Bizer. *Quality-Driven Information Filtering in the Context of Web-Based Information Systems*. VDM Verlag, 2007.
- [16] Woodrow Wilson (Woody) Bledsoe. Non-resolution theorem proving. *Artificial Intelligence*, 9(1):1 – 35, 1977.
- [17] Kirsimarja Blomqvist. The many faces of trust. *Scandinavian Journal of Management*, 13(3):271–286, 1997.
- [18] Matthew Bovee, Rajendra P Srivastava, and Brenda Mak. A conceptual framework and belief-function approach to assessing overall information quality. *International journal of intelligent systems*, 18(1):51–74, 2003.
- [19] Mark A. Cameron, Robert Power, Bella Robinson, and Jie Yin. Emergency situation awareness from twitter for crisis management. In *Proceedings of the 21st international conference companion on World Wide Web, WWW '12 Companion*, pages 695–698, New York, NY, USA, 2012. ACM.
- [20] Thomas Chesney. An empirical examination of Wikipedia’s credibility. *First Monday*, 11, 2006.
- [21] Laurence Cholvy. A General Framework for Reasoning About Contradictory Information and Some of Its Applications. In *Proceedings of the 13th European Conference on Artificial Intelligence*. John Wiley & Sons, Ltd., 1998.
- [22] Eugene Ciurana. *Developing with Google App Engine*. Apress, Berkely, CA, USA, first edition, 2009.
- [23] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. The MIT Press, Cambridge, Massachusetts, 1999. p.3.

- [24] CNN International. Tweeting the terror: How social media reacted to Mumbai. <http://edition.cnn.com/2008/WORLD/asiapcf/11/27/mumbai.twitter/index.html>, 2008. Accessed on 19/04/2010.
- [25] Patrick Collinson. HMRC publishes flickr gallery of 2012's 'top tax criminals'. <http://www.guardian.co.uk/business/2013/jan/04/hmrc-flickr-gallery-tax-criminals>, January 2013. Accessed on 06/01/2013.
- [26] Computerworld.com. [http://www.computerworld.com/s/article/9145259/Facebook\\_creates\\_site\\_dedicated\\_to\\_providing\\_earthquake\\_info](http://www.computerworld.com/s/article/9145259/Facebook_creates_site_dedicated_to_providing_earthquake_info), 2010. Accessed on 15/03/2010.
- [27] Contentious.com. Tracking a rumor: Indian government, twitter, and common sense. <http://www.contentious.com/2008/11/27/tracking-a-rumor-indian-government-twitter-and-common-sens/>, 2008. Accessed on 28/04/2010.
- [28] Rachel Craddock and Steve Saunders. Improved situation awareness through the fusion of tracks from radar and RESM sensors. In *The Use of Radars with Other Sensing Types (Ref. No. 2000/066)*, IEE Colloquium on, pages 3/1–3/3, 2000.
- [29] Sadie Creese. *Data Independent Induction: CSP Model Checking of Arbitrary Sized Networks*. PhD thesis, University of Oxford, UK, 2001.
- [30] Chenyun Dai, Dan Lin, Elisa Bertino, and Murat Kantarcioglu. An approach to evaluate data trustworthiness based on data provenance. In Willem Jonker and Milan Petkovič, editors, *Fifth VLDB Workshop on Secure Data Management (SDM 2008)*, volume 5159 of LNCS, pages 82–98. Springer, 2008.
- [31] Jason Deans, John Plunkett, and Josh Halliday. BBC Weather Twitter account taken over by Syrian regime supporters. <http://www.guardian.co.uk/media/2013/mar/21/bbc-weather-twitter-syrian-regime>, 2013. Accessed on 21/03/2013.
- [32] Oxford Dictionaries. <http://oxforddictionaries.com>, 2012. Accessed on 19/10/2012.
- [33] Oxford Dictionaries. <http://oxforddictionaries.com>, 2012. Accessed on 22/08/2014.
- [34] Cambridge Advanced Learner's Dictionary. <http://dictionary.cambridge.org/dictionary/british/>, 2012. Accessed on 19/10/2012.
- [35] Jin Song Dong, Jing Sun, and Hai Wang. Checking and Reasoning about Semantic Web through Alloy. In *Proceedings of 12th International Symposium on Formal Methods Europe: FM03*, pages 796–813. LNCS, Springer-Verlag, 2003.
- [36] Jin Song Dong, Jun Sun, Hai H. Wang, Chew Hung Lee, and Hian Beng Lee. Analysing Web Ontology in Alloy: A Military Case Study. In *Proceedings of the Fifteenth International Conference on Software Engineering & Knowledge Engineering*, pages 542–546, July 2003.

- [37] Oskar Dressler and Peter Struss. The Consistency-based Approach to Automated Diagnosis of Devices. In Gerhard Brewka, editor, *Principles of Knowledge Representation*, pages 267–311. CSLI Publications, Stanford, 1997.
- [38] Julie Dugdale, Bartel Van de Walle, and Corinna Koeppinghoff. Social media and SMS in the haiti earthquake. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 713–714, New York, NY, USA, 2012. ACM.
- [39] eBay. How Feedback works. <http://pages.ebay.co.uk/help/feedback/howitworks.html>. Accessed on 14/02/2013.
- [40] Amr El Abaddi, Lars Backstrom, Soumen Chakrabarti, Alejandro Jaimes, Jure Leskovec, and Andrew Tomkins. Social media: source of information or bunch of noise. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 327–328, New York, NY, USA, 2011. ACM.
- [41] Mihai Cristian Florea and loi Bossé. Crisis management using Dempster Shafer theory: Using dissimilarity measures to characterize sources' reliability. *C3I for Crisis, Emergency and Consequence Management, (RTO-MP-IST-086)*, May 2009.
- [42] OASIS : Open Advanced System for dIsaster and emergency management. [http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ\\_RCN=10981694](http://cordis.europa.eu/search/index.cfm?fuseaction=proj.document&PJ_RCN=10981694), 2008. Accessed on 02/11/2012.
- [43] European Committee for Standardization (CEN). Disaster and emergency management - Shared situation awareness - Part 1: Message structure, CEN Workshop Agreement (CWA 15931-1). [http://www.tacticalsituationobject.org/docs/CWA\\_15931-1.pdf](http://www.tacticalsituationobject.org/docs/CWA_15931-1.pdf), 2009. Accessed on 23/11/2012.
- [44] European Committee for Standardization (CEN). Disaster and emergency management - Shared situation awareness - Part 2: Codes for the message structure, CEN Workshop Agreement (CWA 15931-2). [http://www.tacticalsituationobject.org/docs/CWA\\_15931-2.pdf](http://www.tacticalsituationobject.org/docs/CWA_15931-2.pdf), 2009. Accessed on 23/11/2012.
- [45] freshdesk.com. Improving Customer Service with Twitter. <http://freshdesk.com/whitepapers/twitter-monitoring-using-freshdesk.pdf>. Accessed on 02/01/2013.
- [46] Karen K. Fullam and K. Suzanne Barber. Using policies for information valuation to justify beliefs. In *Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS'04)*, July 2004.
- [47] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S. Yu, and Hongjun Lu. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, pages 181–192. VLDB Endowment, 2005.
- [48] Claire Gardent and Bonnie Webber. Towards the Use of Automated Reasoning in Discourse Disambiguation. *J. of Logic, Lang. and Inf.*, 10(4):487–509, September 2001.

- [49] Yolanda Gil and Donovan Artz. Towards content trust of web resources. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 5(4):227–239, December 2007.
- [50] Tyrone Grandison and Morris Sloman. A survey of trust in internet applications. *IEEE Communications Surveys and Tutorials*, pages 2–16, Fourth Quarter 2000.
- [51] The Stanford Natural Language Processing Group. Stanford Named Entity Recognizer (NER). <http://nlp.stanford.edu/software/CRF-NER.shtml>. Accessed on 01/01/2013.
- [52] BBC admits it made mistakes using Mumbai Twitter coverage. <http://www.guardian.co.uk/media/pda/2008/dec/05/bbc-twitter>, November 2008. Accessed on 21/04/2010.
- [53] Sanjeev Gupta and Mayank Dave. Cellular distribution framework for wireless sensor based decision support systems for crisis management model. *Indian Journal of Science and Technology*, 2(11):41–45, November 2009.
- [54] Robert A. Hackett and Yuezhi Zhao. *Sustaining democracy? : journalism and the politics of objectivity*. Garamond Press, Toronto, 1998.
- [55] Joseph Y. Halpern and Moshe Y. Vardi. Model checking vs. theorem proving: a manifesto. In Vladimir Lifschitz, editor, *Artificial intelligence and mathematical theory of computation*, pages 151–176. Academic Press Professional, Inc., San Diego, CA, USA, 1991.
- [56] Olaf Hartig. Provenance information in the web of data. In *Linked Data on The Web (LDOW2009)*, April 2009. [http://ceur-ws.org/Vol-538/ldow2009\\_paper18.pdf](http://ceur-ws.org/Vol-538/ldow2009_paper18.pdf).
- [57] Olaf Hartig and Jun Zhao. Using web data provenance for quality assessment. In *First International Workshop on the Role of Semantic Web in Provenance Management (SWPM'09)*, 2009.
- [58] Qingping He. Estimating the Reliability of Composite Scores. <http://dera.ioe.ac.uk/id/eprint/1060>, 2009. Accessed on 08/08/2012.
- [59] James Heather and Kun Wei. Where Next for Formal Methods? In Bruce Christianson, Bruno Crispo, JamesA. Malcolm, and Michael Roe, editors, *Security Protocols*, volume 5087 of *Lecture Notes in Computer Science*, pages 52–58. Springer Berlin Heidelberg, 2009.
- [60] Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H. Chi. Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 237–246, New York, NY, USA, 2011. ACM.
- [61] Anthony Hunter. How to act on inconsistent news: ignore, resolve, or reject. *Data Knowl. Eng.*, 57(3):221–239, June 2006.

- [62] Internet Engineering Task Force (IETF). The JavaScript Object Notation (JSON) Data Interchange Format. <http://tools.ietf.org/html/rfc7159>, 2014. Accessed on 22/08/2014.
- [63] Oxfam International. Haiti earthquake 2010. <http://www.oxfam.org/en/haitiquake>. Accessed on 07/01/2013.
- [64] Danial Jackson. *Software Abstractions: Logic, Language, and Analysis*. MIT Press, 2006. p.83.
- [65] Danial Jackson. *Software Abstractions: Logic, Language, and Analysis*. MIT Press, 2006. p.150.
- [66] Predrag Janicic. Automated Reasoning: Some Successes and New Challenges. In *Proceedings of the 22nd Central European Conference on Information and Intelligent Systems (CECIS 2011)*, Varaždin, Croatia, September 21-23 2011.
- [67] Maya Nand Jha, Jason Levy, and Yang Gao. Advances in Remote Sensing for Oil Spill Disaster Management: State-of-the-Art Sensors Technology for Oil Spill Surveillance. *Sensors*, 8(1):236–255, 2008.
- [68] Fredrik Johansson, Joel Brynielsson, Pontus Horling, Michael Malm, Christian Martenson, Staffan Truvé, and Magnus Rosell. Detecting Emergent Conflicts through Web Mining and Visualization. In *Intelligence and Security Informatics Conference (EISIC), 2011 European*, pages 346–353, September 2011.
- [69] Kimberly Keeton, Pankaj Mehra, and John Wilkes. Do you know your IQ?: a research agenda for information quality in systems. *ACM SIGMETRICS Performance Evaluation Review*, 37(3):26–31, 2010.
- [70] Kari Kelton, Kenneth R. Fleischmann, and William A. Wallace. Trust in digital information. *Journal of the American Society for Information Science and Technology*, 59(3):363–374, 2008.
- [71] Ashish Khetan. 60 dark hours at Hotel Taj. In Harinder Baweja, editor, *26/11 Mumbai attacked*, pages 46–83. Roli Books, 2009.
- [72] Shirlee-Ann Knight and Janice Bum. Developing a framework for assessing information quality on the world wide web. *Informing Science: International Journal of an Emerging Transdiscipline*, 8(5):159–172, 2005.
- [73] Mayank Lahiri. Gaming foursquare with 9 lines of Perl. <http://compbio.cs.uic.edu/~mayank/4sq.html>, August 2010. Retrieved on 13/03/2011.
- [74] Amy N. Langville and Carl D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–380, 2004.
- [75] Algirdas Laukaitis and Olegas Vasilecas. Automatic formal verification of conceptual model documentation by means of self-organizing map. *Advances in Computational Linguistics, a special issue of Research in Computing Science*, 41:143–154, 2009.

- [76] Thierry Lecomte, Lilian Burdy, and Michael Leuschel. Formally Checking Large Data Sets in the Railways. *CoRR*, abs/1210.6815, 2012.
- [77] John David Lewis and Andrew Weigert. Trust as a social reality. *Social Forces*, 63(4):967–985, 1985.
- [78] Gregory R. Madey, Albert-László Barabási, Nitesh V. Chawla, Marta Gonzalez, David Hachen, Brett Lantz, Alec Pawling, Timothy Schoenharl, Gábor Szabó, Pu Wang, and Ping Yan. Enhanced Situational Awareness: Application of DDDAS Concepts to Emergency and Disaster Management. In *Proceedings of the 7th international conference on Computational Science, Part I: ICCS 2007*, ICCS '07, pages 1090–1097, Berlin, Heidelberg, 2007. Springer-Verlag.
- [79] Marco Manso and Brbara Manso. The Role of Social Media in Crisis: A Holistic Approach to European Adoption of the Online and Mobile Communications in Crisis Response and Search and Rescue Efforts. In *International Command and Control Research and Technology Symposium, ICCRTS, WWW '12 Companion*, 2012.
- [80] Andrew Martin, John Lyle, and Cornelius Namilkuo. Provenance as a security control. In *Proceedings of the 4th USENIX conference on Theory and Practice of Provenance*, TaPP'12, pages 3–3, Berkeley, CA, USA, 2012. USENIX Association.
- [81] David Maxwell, Stefan Raue, Leif Azzopardi, Chris Johnson, and Sarah Oates. Crisees: Real-Time Monitoring of Social Media Streams to Support Crisis Management. In Ricardo Baeza-Yates, Arjen P. Vries, Hugo Zaragoza, B.Barla Cambazoglu, Vanessa Murdock, Ronny Lempel, and Fabrizio Silvestri, editors, *Advances in Information Retrieval*, volume 7224 of *Lecture Notes in Computer Science*, pages 573–575. Springer Berlin Heidelberg, 2012.
- [82] Declan McCullagh. FBI renews broad internet surveillance push. [http://news.bbc.co.uk/1/hi/world/south\\_asia/7752003.stm](http://news.bbc.co.uk/1/hi/world/south_asia/7752003.stm), September 2012. Accessed on 20/10/2012.
- [83] P. McDaniel. Data Provenance and Security. *Security Privacy, IEEE*, 9(2):83–85, March-April 2011.
- [84] J.D. Meier, Srinath Vasireddy, Ashish Babbar, and Alex Mackman. *Improving .NET Application Performance and Scalability*. Microsoft Corporation, May 2004.
- [85] Patrick Meier. Crowdsourcing and Validation. <http://www.understandrisk.org/ur/node/4596>, March 2010. Accessed on 27/09/2012.
- [86] Nasrullah Memon, David L. Hicks, and Henrik Legind Larsen. Harvesting Terrorists Information from Web. In *Proceedings of the 11th International Conference Information Visualization, IV '07*, pages 664–671, Washington, DC, USA, 2007. IEEE Computer Society.
- [87] Marcelo Mendoza, Barbara Poblete, and Carlos Castillo. Twitter under crisis: can we trust what we RT? In *Proceedings of the First Workshop on Social Media Analytics, SOMA '10*, pages 71–79, New York, NY, USA, 2010. ACM.

- [88] Raina M. Merchant, Stacy Elmer, and Nicole Lurie. Integrating Social Media into Emergency-Preparedness Efforts. *New England Journal of Medicine*, 365(4):289–291, 2011.
- [89] Panagiotis Takis Metaxas and Eni Mustafaraj. Trails of Trustworthiness in Real-Time Streams. In *Proceedings of the 15th ACM Conference on Computer Supported Cooperative Work (CSCW 2012)*. ACM, February 2012.
- [90] Miriam J. Metzger. Making sense of credibility on the Web: Models for evaluating online information and recommendations for future research. *J. Am. Soc. Inf. Sci. Technol.*, 58(13):2078–2091, November 2007.
- [91] Stanley Milgram. The Small World Problem. *Psychology Today*, 2:60–67, 1967.
- [92] MIT Software Design Group. Alloy: A Language & Tool for Relational Models. <http://alloy.mit.edu>, 2011. Accessed on 25/05/2012.
- [93] Microsoft Developer Network (MSDN). Code Metrics Values. [https://msdn.microsoft.com/en-us/library/bb385914\(v=vs.100\).aspx](https://msdn.microsoft.com/en-us/library/bb385914(v=vs.100).aspx). Accessed on 16/03/2015.
- [94] Victor Naroditskiy, Iyad Rahwan, Manuel Cebrian, and Nicholas R. Jennings. Verification in referral-based crowdsourcing. *PLoS ONE*, 7(10):e45924, October 2012.
- [95] [http://nationnewmedia.com/2010/01/17/opinion/opinion\\_30120439.php](http://nationnewmedia.com/2010/01/17/opinion/opinion_30120439.php), 2010. Accessed on 15/03/2010.
- [96] The European Network and Information Security Agency (ENISA). Web 2.0 survey. [http://www.enisa.europa.eu/act/it/oar/web-2.0-security-and-privacy/enisa\\_survey\\_web2.pdf/view](http://www.enisa.europa.eu/act/it/oar/web-2.0-security-and-privacy/enisa_survey_web2.pdf/view). Accessed on 20/11/2009.
- [97] The European Network and Information Security Agency (ENISA). Web 2.0 security and privacy. <http://www.enisa.europa.eu/act/it/oar/web2sec/report>, December 2008.
- [98] BBC News. Haiti will not die, president rene preval insists. <http://news.bbc.co.uk/1/hi/world/americas/8511997.stm>, 2010. Accessed on 07/01/2013.
- [99] BBC News. AP Twitter account hacked in fake ‘White House blasts’ post. <http://www.bbc.co.uk/news/world-us-canada-21508660>, 2013. Accessed on 02/05/2013.
- [100] Yahoo! News. Tax Cheats: HMRC Publishes Names and Photos. <http://uk.news.yahoo.com/tax-cheats-hmrc-publishes-names-photos-130854787.html>, January 2013. Accessed on 06/01/2013.
- [101] Nigel Nicholson, Emma Soane, Mark Fenton-O’Creevy, and Paul Willman. Personality and domain-specific risk taking. *Journal of Risk Research*, 8(2):157–176, 2005.

- [102] Jason R.C. Nurse, Sadie Creese, Michael Goldsmith, Syed Sadiqur Rahman, David Lund, George Mourikas, Darren Price, and T. Baker. Consolidated report on trustworthiness measures and overlay design. Technical Report TEASE D3.6 Issue 1.0, TEASE Consortium, UK, February 2013.
- [103] Jason R.C. Nurse, Syed Sadiqur Rahman, Sadie Creese, Michael Goldsmith, and Koen Lamberts. Information Quality and Trustworthiness: A Topical State-of-the-Art Review. In *The International Conference on Computer Applications and Network Security (ICCANS) 2011*. IEEE, 2011.
- [104] Press Information Bureau (Government of India). HM announces measures to enhance security. <http://pib.nic.in/release/release.asp?relid=45446>, December 2008. Press release. Retrieved on 07/05/2010.
- [105] Onook Oh, Manish Agrawal, and H. Raghav Rao. Information control and terrorism: Tracking the Mumbai terrorist attack through twitter. *Information Systems Frontiers*, 13(1):33–43, March 2011.
- [106] Martin Ouimet. Formal Software Verification: Model Checking and Theorem Proving. Embedded systems laboratory technical report, esl-tik-00214, Embedded Systems Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA, 2008.
- [107] Jeff Pasternack and Dan Roth. Comprehensive Trust Metrics for Information Networks. In *ASC*, Orlando, Florida, 12 2010.
- [108] Leo L Pipino, Yang W Lee, and Richard Y Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, 2002.
- [109] KEVIN POULSEN. Firsthand Reports From California Wildfires Pour Through Twitter. <http://www.wired.com/threatlevel/2007/10/firsthand-repor/>, October 2007. Accessed on 16/11/2012.
- [110] Syed Sadiqur Rahman. Phishing attack. Master’s thesis, University of East London, 2006.
- [111] Syed Sadiqur Rahman, Sadie Creese, and Michael Goldsmith. Accepting Information with a Pinch of Salt: Handling Untrusted Information Sources. In Catherine Meadows and Carmen Fernandez-Gago, editors, *Security and Trust Management*, volume 7170 of *Lecture Notes in Computer Science*, pages 223–238. Springer Berlin Heidelberg, 2012.
- [112] MIT Media Relations. Media Lab team wins DARPA’s Red Balloon Challenge. <http://web.mit.edu/press/2009/darpa-challenge-1210.html>, December 2009. Accessed on 24/10/2012.
- [113] Swift River. <https://swiftapp.com/welcome>. Accessed on 02/11/2012.
- [114] Julian B. Rotter. A new scale for the measurement of interpersonal trust<sup>1</sup>. *Journal of Personality*, 35(4):651–665, 1967.



- [115] Julian B. Rotter. Interpersonal Trust, Trustworthiness, and Gullibility. *American Psychologist*, 35(1):1–7, 1980.
- [116] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM.
- [117] Erick Schonfeld. Costolo: Twitter Now Has 190 Million Users Tweeting 65 Million Times A Day. <http://techcrunch.com/2010/06/08/twitter-190-million-users>, July 2010. Accessed on 10/12/2012.
- [118] Axel Schulz, Heiko Paulheim, and Florian Probst. Crisis Information Management in the Web 3.0 Age. In *9th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, 2012.
- [119] Michael L. Scott. *Programming Language Pragmatics*. Morgan Kaufman, Department of Computer Science University of Rochester, second edition, 2006.
- [120] Len Seligman, Barbara Blaustein, Peter Mork, Ken Smith, and Neal Rothleder. Information interoperability and provenance for emergency preparedness and response. In *Technologies for Homeland Security, 2009. HST '09. IEEE Conference on*, pages 529–536, May 2009.
- [121] Aladdin Shamoug and Radmila Juric. Ontological Reasoning as a Tool for Humanitarian Decision Making. In *Proceedings of the 9th International Conference on Information Systems for Crisis Response and Management*. ISCRAM, April 2012.
- [122] Graeme Shanks and Brian Corbitt. Understanding data quality: Social and cultural aspects. In *Proceedings of the 10th Australasian Conference on Information Systems*, volume 785. Citeseer, 1999.
- [123] Blair H. Sheppard and Dana M. Sherman. The grammars of trust: A model and general implications. *Academy of Management Review*, 23(3):422–437, July 1998.
- [124] Junsuk Shin, Rajnish Kumar, Dushmanta Mohapatra, Umakishore Ramachandran, and Mostafa Ammar. ASAP: a camera sensor network for situation awareness. In *Proceedings of the 11th international conference on Principles of distributed systems*, OPODIS'07, pages 31–47, Berlin, Heidelberg, 2007. Springer-Verlag.
- [125] MG Siegler. Twitter quakes as San Francisco shakes. <http://venturebeat.com/2009/03/30/twitter-quakes-as-san-francisco-shakes/>, March 2009. Accessed on 16/11/2012.
- [126] Matt SilvermanMar. A Day in the Life of the Internet. <http://mashable.com/2012/03/06/one-day-internet-data-traffic/>, 2012. Accessed on 09/04/2013.
- [127] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance in e-science. *SIGMOD Rec.*, 34(3):31–36, September 2005.

- [128] Jocelyn Simmonds and M. Cecilia Bastarrica. A tool for automatic UML model consistency checking. In *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering, ASE '05*, pages 431–432, New York, NY, USA, 2005. ACM.
- [129] Carsten Sinz, Andreas Kaiser, and Wolfgang Kuchlin. SAT-based consistency checking of automotive electronic product data. In *Configuration Workshop Proceedings, 14th European Conference on Artificial Intelligence (ECAI-2000)*, pages 74–78, Berlin, Germany, August 2000.
- [130] Carsten Sinz, Andreas Kaiser, and Wolfgang Kuchlin. Detection of Inconsistencies in Complex Product Model Data Using Extended Propositional SAT-Checking. In I. Russell and J. Kolen, editors, *Proceedings of the 14th International FLAIRS Conference*, pages 645–649, Key West, FL, MAY 2001. AAAI Press.
- [131] Paul R. Smart, Alistair Russell, Nigel R. Shadbolt, M. C. Shraefel, and Leslie A. Carr. AKTiveSA: a technical demonstrator system for enhanced situation awareness. *THE COMPUTER JOURNAL*, 50(6):703–716, November 2007.
- [132] Paul R Smart, Nigel R Shadbolt, Leslie A Carr, and monica c schraefel. Knowledge-Based Information Fusion for Improved Situational Awareness. In *8th International Conference on Information Fusion*, 2005. Event Dates: 25-29 July 2005.
- [133] SMS packet format. <http://www.gsmfavorites.com/documents/sms/packetformat/>. Accessed on 13/03/2011.
- [134] Kim Stephens. Swift River: A tool for organizing Social Media Crisis Information. <http://idisaster.wordpress.com/2011/01/03/swift-river-a-tool-for-organizing-social-media-crisis-information/>, January 2011. Accessed on 27/09/2012.
- [135] Natalie Jomini Stroud and Stephen D. Reese. Objectivity & Balance: How Do Readers and Viewers of News and Information Reach Conclusions Regarding Objectivity and Balance? White Paper, <http://www.cpb.org/aboutcpb/goals/objectivity/whitepapers/>, October 2007.
- [136] Tomek Strzalkowski, Fang Lin, Jin Wang, and Jose Perez-Carballo. Evaluating natural language processing techniques in information retrieval. In *Natural language information retrieval*, pages 113–145. Springer, 1999.
- [137] Thanwadee Sunetnanta and Anthony Finkelstein. Automated Consistency Checking for Multiperspective Software Specifications. In Peri Tarr and Harold Ossher, editors, *Workshop on Advanced Separation of Concerns in Software Engineering (ICSE 2001)*, May 2001.
- [138] L.M. Surhone, M.T. Tennoe, and S.F. Henssonow. *Advanced Persistent Threat*. VDM Publishing, 2010.
- [139] Matthias Suttera and Martin G. Kocher. Trust and trustworthiness across different age groups. *Games and Economic Behavior*, 59(2):364–382, May 2007.

- [140] Wang-Chiew Tan. Provenance in databases: Past, current, and future. *IEEE Data Engineering Bulletin*, 30(4):3–12, December 2007.
- [141] John C. Tang, Manuel Cebrian, Nicklaus A. Giacobe, Hyun-Woo Kim, Taemie Kim, and Douglas "Beaker" Wickert. Reflecting on the DARPA Red Balloon Challenge. *Commun. ACM*, 54(4):78–85, April 2011.
- [142] Colin Tankard. Advanced persistent threats and how to monitor and deter them. *Network Security*, 2011(8):16 – 19, 2011.
- [143] Andrea H. Tapia, Kartikeya Bajpai, Jim Jansen, John Yen, and Lee Giles. Seeking the Trustworthy Tweet: Can Microblogged Data Fit the Information Needs of Disaster Response and Humanitarian Relief Organizations. In *Proceedings of the 8th International Conference on Information Systems for Crisis Response and Management*. ISCRAM, 05 2011.
- [144] Mumbai attacks: Twitter and Flickr used to break news. <http://www.telegraph.co.uk/news/worldnews/asia/india/3530640/Mumbai-attacks-Twitter-and-Flickr-used-to-break-news-Bombay-India.html>, November 2008. Accessed on 14/02/2014.
- [145] Daniel Terdiman. Report: Twitter hits half a billion tweets a day. [http://news.cnet.com/8301-1023\\_3-57541566-93/report-twitter-hits-half-a-billion-tweets-a-day/](http://news.cnet.com/8301-1023_3-57541566-93/report-twitter-hits-half-a-billion-tweets-a-day/). Accessed on 04/01/2013.
- [146] Robert Thomson, Naoya Ito, Hinako Suda, Fangyu Lin, Yafei Liu, Ryo Hayasaka, Ryuzo Isochi, and Zian Wang. Trusting Tweets: The Fukushima Disaster and Information Source Credibility on Twitter. In *Proceedings of the 9th International Conference on Information Systems for Crisis Response and Management*. ISCRAM, April 2012.
- [147] Twitter. GET search (API version 1). <https://dev.twitter.com/docs/api/1/get/search>. Accessed on 14/02/2014.
- [148] Twitter. GET statuses/user\_timeline. [https://dev.twitter.com/docs/api/1/get/statuses/user\\_timeline](https://dev.twitter.com/docs/api/1/get/statuses/user_timeline). Accessed on 04/01/2013.
- [149] Twitter. Using the Twitter Search API. <https://dev.twitter.com/docs/using-search>. Accessed on 04/01/2013.
- [150] Denise Umuhoza and Robin Braun. Trustworthiness Assessment of Knowledge on the Semantic Sensor Web by Provenance Integration. In *Proceedings of the 2012 26th International Conference on Advanced Information Networking and Applications Workshops*, WAINA '12, pages 387–392, Washington, DC, USA, 2012. IEEE Computer Society.
- [151] W3C Markup Validation Service. RSS 2.0 specification. <http://validator.w3.org/feed/docs/rss2.html>, 2008. Accessed on 20/05/2010.

- [152] Hai H. Wang, Jin Song Dong, Jing Sun, and Jun Sun. Reasoning support for semantic web ontology family languages using alloy. In *International Journal of Multiagent and Grid Systems, Special issue on Agent-Oriented Software Development Methodologies*, 2006.
- [153] Richard Y Wang and Diane M Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, pages 5–33, 1996.
- [154] Xinlei Wang, K. Govindan, and P. Mohapatra. Provenance-Based Information Trustworthiness Evaluation in Multi-Hop Networks. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1 –5, December 2010.
- [155] Xinlei (Oscar) Wang, Kannan Govindan, and Prasant Mohapatra. Collusion-resilient quality of information evaluation based on information provenance. In *SECON*, pages 395–403, 2011.
- [156] Zhiyi Wang, Matti Hämäläinen, and Zhangxi Lin. An Open Community Approach to Emergency Information Services during a Disaster. In *Proceedings of the 2008 International Symposium on Information Science and Engineering - Volume 01, ISISE '08*, pages 649–654, Washington, DC, USA, 2008. IEEE Computer Society.
- [157] David H. D. Warren. Implementing Prolog – compiling predicate logic programs. D.A.I Research Report 39 and 40, University of Edinburgh, Edinburgh, Scotland, 1977.
- [158] Elke U Weber, Ann-Renée Blais, and Nancy E Betz. A domain-specific risk-attitude scale: Measuring risk perceptions and risk behaviors. *Journal of behavioral decision making*, 15(4):263–290, 2002.
- [159] Wikipedia. [http://simple.wikipedia.org/wiki/November\\_2008\\_Mumbai\\_attacks](http://simple.wikipedia.org/wiki/November_2008_Mumbai_attacks). Accessed on 19/04/2010.
- [160] Leslie B. Wilson and Robert George Clark. *Comparative programming languages*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1988.
- [161] Shouhuai Xu, Ravi Sandhu, and Elisa Bertino. TIUPAM: A framework for trustworthiness-centric information sharing. In Ferrari, editor, *Third IFIP International Conference on Trust Management (TM'09)*, volume 300 of *IFIP AICT*, pages 164–175, 2009.
- [162] Nisansala Yatapanage. *Slicing Behavior Trees for Verification of Large Systems*. PhD thesis, Griffith University, Australia, June 2012).
- [163] Nisansala Yatapanage, Kirsten Winter, and Saad Zafar. Slicing behavior tree models for verification. In Cristian S. Calude and Vladimiro Sassone, editors, *Theoretical Computer Science*, volume 323 of *IFIP Advances in Information and Communication Technology*, pages 125–139. Springer Berlin Heidelberg, 2010.
- [164] Jie Yin, Sarvnaz Karimi, Bella Robinson, and Mark Cameron. ESA: emergency situation awareness via microbloggers. In *Proceedings of the 21st ACM international*

- conference on Information and knowledge management, CIKM '12*, pages 2701–2703, New York, NY, USA, 2012. ACM.
- [165] Xiaoxin Yin, Jiawei Han, and Philip S. Yu. Truth discovery with multiple conflicting information providers on the web. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 1048–1052, New York, NY, USA, 2007. ACM.
- [166] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.
- [167] Xiuqin Zhong, Hongguang Fu, and Yan Jiang. Coupling ontology with rule-based theorem proving for knowledge representation and reasoning. In Yanchun Zhang, Alfredo Cuzzocrea, Jianhua Ma, Kyo-il Chung, Tughrul Arslan, and Xiaofeng Song, editors, *Database Theory and Application, Bio-Science and Bio-Technology*, volume 118 of *Communications in Computer and Information Science*, pages 110–119. Springer Berlin Heidelberg, 2010.
- [168] Jan Zibuschka, Uwe Laufs, and Heiko Ronagel. Towards Ubiquitous Emergency Management Systems. In *Proceedings of the 1st International Workshop on Model-based Interactive Ubiquitous Systems*, pages 21–26, June 2011.
- [169] Randi Zuckerberg. Global Relief on Facebook: Connecting in Times of Need. <http://blog.facebook.com/blog.php?post=252151582130>, 2010. Accessed on 07/01/2013.

# Glossary

## **Advanced Persistent Threat**

An advanced persistent threat is an attack on the information system, in which an unauthorized agent gains access to a system and stays undetected for a prolonged duration [138].

## **Alloy**

Alloy is a model specification language based on first-order logic and set theory [92]. Alloy specifies objects (e.g. person, vehicle, etc.), properties of objects (e.g. *name* of a person, *make* and *model* of a vehicle) and their relations (relations between objects and relations between an object and its properties). An Alloy model also includes constraints and expressions (Facts, Predicates, Assertions, etc.) [64].

## **Alloy Analyzer**

The Alloy Analyzer [92, 64], is a constraint solver or model finder that finds a model by analysing specifications written in the Alloy specification language. Alloy Analyzer takes the constraints of a model and tries to find an instance that satisfies them within a given scope (see Section 7.2.3 for more details on *scope*). Alloy Analyzer internally uses a SAT solver to check if a predicate is satisfiable and returns true or false based on the predicate's satisfiability.

## **API**

API stands for Application Programming Interface. An API specifies how different components of software can interact with each other. For example, an API specifies the types and data structures of the input and output of a software component.

## **Atom or Atom Syndication Format**

The Atom Syndication Format is an XML-based web content and metadata syndication format that describes a list of related information known as ‘feed’.

### **Boolean Satisfiability**

SAT stands for ‘SATisfiability’ or Boolean Satisfiability [65]. A Boolean formula is an expression, which is composed of one or more Boolean variables (that can take either of the two values: True and False) joined by Boolean operators (AND, OR, NOT). Thus a Boolean expression can only evaluate to True or False. A formula is called satisfiable, if it can be made True by assigning appropriate values (True or False) to its variables. Boolean satisfiability problem (SAT) is the problem of determining whether there is any interpretation of the variables in a given Boolean formula, which satisfies the formula (i.e. the assignment or interpretation makes the expression True).

### **Boolean Satisfiability Solver**

A Boolean Satisfiability Solver or SAT Solver is a software application that evaluates whether a Boolean expression is satisfiable or not. In other words, a SAT Solver tries to find a suitable assignment of values to the formula’s boolean variables that satisfies the formula (i.e. makes the formula True).

### **Crowdsourcing**

Crowdsourcing is the practice of obtaining required information or services by enlisting the services of a number of people, either paid or unpaid, typically via the Internet [32].

### **Geocoding**

Geocoding is the process of finding the geographic coordinates (latitude and longitude) of a given location, such as an address, post code, the name of a street or place of interest.

### **JSON**

JSON stands for JavaScript Object Notation [62]. JSON is a lightweight, text-based, language-independent data interchange format. It defines a small set of formatting rules for the portable representation of structured data [62].

### **Phishing**

Phishing is a form of online identity theft [110]. Phishing is the fraudulent practice of sending fake emails, or spam, purporting to be from banks or other trusted organisations, in an attempt to fool the user into disclosing sensitive

information such as username, password, account details, ATM Pin or credit card details [110, 33].

### **Provenance of Information**

Provenance (also referred to as lineage and pedigree) of information refers to the source of information such as who gave (or produced) the information, the derivation history of information, what data was used to generate it, finding the trail of how the information has passed from one source to the other, and how it has been changed.

### **RSS**

RSS stands for Really Simple Syndication [151]. RSS is a Web content syndication format, which helps delivering frequently changing web content. RSS feeds enable online publishers (e.g. news and weather services) to syndicate their updates (information) automatically.

### **SAT Solver**

SAT stands for ‘SATisfiability’ or Boolean Satisfiability [64]. Hence, the term ‘SAT Solver’ refers to a Boolean Satisfiability Solver. (See the definition of Boolean Satisfiability Solver for more details.)

### **SQL**

SQL stands for Structured Query Language. SQL is a database query language, which is designed for managing data stored in a relational database management system such as Oracle, Microsoft SQL Server, DB2, etc.

### **TSO**

The Tactical Situation Object (TSO) is an object containing language-independent situation information encoded in XML, following the format designed and proposed by OASIS [42]. The TSO can be used to describe an event or incident, the resources available or deployed in the operation after the incident, and the missions in progress.

### **World View**

A world view, which is collection of consistent messages harvested from open-sources, provides a possible picture of a situation constructed from the evidence found in the messages.



## **XML**

XML stands for eXtensible Markup Language. So, XML is a markup language that defines a set of rules for encoding documents in a simple and flexible text format. XML is very useful in the exchange of a wide variety of data on the Web and elsewhere.

## **YAML**

YAML is the abbreviation for ‘YAML Ain’t Markup Language’. Notice that the acronym ‘YAML’ is also a part of its unabbreviated name. YAML is a data serialization standard for programming languages. YAML is easier to read and requires less runtime resources than XML [22].

# Appendices

## APPENDIX A

# Structure of Tweets Returned by the Twitter APIs

Twitter provides a number of APIs to meet different requirements of the application developers who use Twitter data. Each API requires a different set of input parameters and returns different output in different formats. As mentioned in Chapter 3, I have used two Twitter APIs, namely *User Timeline API* and *Search API*.

### A.1 Structure of Tweets Returned by the Twitter User Timeline API

The following tweet shows the structure of the tweets in XML format returned by the *Twitter User Timeline API*. This tweet was made by *Dina*, one of the Twitter users, on Thursday 27 November 2008 (at 06:43:47 +0000).

```
<status>
  <created_at>Thu Nov 27 06:43:47 +0000 2008</created_at>
  <id>1025980128</id>
  <text>
    #mumbai Times Now quotes British high commission saying 7 British citizens injured in attacks.
    Israelis being held in & near Chabad House
  </text>
  <source>&lt;a href="http://www.twhirl.org/" rel="nofollow"&gt;twhirl&lt;/a&gt;</source>
  <truncated>>false</truncated>
  <in_reply_to_status_id>
</in_reply_to_status_id>
  <in_reply_to_user_id>
</in_reply_to_user_id>
  <favorited>>false</favorited>
  <in_reply_to_screen_name>
</in_reply_to_screen_name>
  <user>
```

```

<id>11778</id>
<name>Dina Mehta</name>
<screen_name>dina</screen_name>
<location>Mumbai, India</location>
<description>
    Looking outside. Researcher, Ethnographer, Social media observer. http://mosoci.com/
</description>
<profile_image_url>
    http://a1.twimg.com/profile\_images/390799570/dtwitterprofile\_normal.jpg
</profile_image_url>
<url>http://dinamehta.com/</url>
<protected>>false</protected>
<followers_count>6107</followers_count>
<profile_background_color>9ae4e8</profile_background_color>
<profile_text_color>000000</profile_text_color>
<profile_link_color>0000ff</profile_link_color>
<profile_sidebar_fill_color>e0ff92</profile_sidebar_fill_color>
<profile_sidebar_border_color>87bc44</profile_sidebar_border_color>
<friends_count>814</friends_count>
<created_at>Wed Nov 08 06:20:10 +0000 2006</created_at>
<favourites_count>68</favourites_count>
<utc_offset>19800</utc_offset>
<time_zone>Mumbai</time_zone>
<profile_background_image_url>
    http://a1.twimg.com/profile\_background\_images/1071022/postcard\_curves.jpg
</profile_background_image_url>
<profile_background_tile>>true</profile_background_tile>
<notifications>
</notifications>
<geo_enabled>>true</geo_enabled>
<verified>>false</verified>
<following>
</following>
<statuses_count>3320</statuses_count>
<lang>en</lang>
<contributors_enabled>>false</contributors_enabled>
</user>
<geo />
<coordinates />
<place />
<contributors />
</status>

```

## A.2 Structure of Tweets Returned by the Twitter Search API

```
{
  completed_in: 0.039,
  max_id: 327882069613096960,
  max_id_str: "327882069613096960",
  next_page: "?page=16&max_id=327882069613096960&q=%28M62%20OR%20motorway%20OR%20Pontefract%29%20%28Minibus%20OR%20Lorry%20OR%20fatalities%20OR%20injured%20OR%20Ambulance%29\"",
  previous_page: "?page=14&max_id=327882069613096960&q=%28M62%20OR%20motorway%20OR%20Pontefract%29%20%28Minibus%20OR%20Lorry%20OR%20fatalities%20OR%20injured%20OR%20Ambulance%29\"",
  page: 15,
  query: "%28M62+OR+motorwav+OR+Pontefract%29+AND+%28Crash+OR+Collision+OR+Accident%29+AND+%28Minibus+OR+Lorry+OR+fatalities+OR+injured+OR+Ambulance%29\"",
  refresh_url: "?since_id=327882069613096960&q=%28M62%20OR%20motorway%20OR%20Pontefract%29%20AND%20%28Minibus%20OR%20Lorry%20OR%20fatalities%20OR%20injured%20OR%20Ambulance%29\"",
  results: [
    {
      created_at: "Fri, 26 Apr 2013 18:18:31 +0000",
      from_user: "n43neen",
      from_user_id: 351152816,
      from_user_id_str: "351152816",
      from_user_name: "Nazneen",
      geo: null,
      id: 327849183086247940,
      id_str: "327849183086247936",
      iso_language_code: "en",
      metadata: {
        result_type: "recent"
      },
      profile_image_url: http://a0.twimg.com/profile\_images/2972226739/ea8bc21775cc72e433ef7e,
      profile_image_url_https: https://si0.twimg.com/profile\_images/2972226739/ea8bc21775cc72e433ef7e,
      source: "<a href="http://twitter.com/download/iphone">Twitter for iPhone",
      text: "RT @BBCNews: Crash involving minibus, lorry & car closes #M62 - @YorksAmbulance activates \"major incident plan\" http://t.co/dNekwgPHR4"
    },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... },
    { ... }
  ],
  results_per_page: 15,
  since_id: 0,
  since_id_str: "0"
}
```

Figure A.1: Twitter Search API's in Output JSON Format

## APPENDIX B

# Construction of World Views: Input, Process and Output

All of the messages that were used as an input to the Decision Support System for generating world views in Chapter 9 is given below in Table B.1. While a detailed discussion of the process of finding a world view is presented in 7 and the process diagram of the breadth-first search for a world view is shown in Figure 7.15, the process diagram of the depth-first search for a world view is given below in Figure B.1.

### B.1 Input Messages used to Construct World views in Chapter 9

Table B.1: Complete Set of Messages Used to Construct World Views and their Satisfiability Evaluated by the DSS

ID	Proposition	Satisfiable	Scores		
			Location	Freshness	Reputation
A	Blue Nissan hit another Red Car	Yes	4	4	5
B	Car with Reg. No. AB11_XYZ hit a Red BMW on Oxford Street	Yes	3	1	8

*Continued on next page*

Table B.1 – *Continued from previous page*

ID	Proposition	Satisfiable	Scores		
			Location	Freshness	Reputation
C	Green Car with Reg AB11_XYZ hit Red BMW on Oxford Street	No	5	7	-2
D	Blue Car with Reg. No. AB11_XYZ hit a Red Car on Oxford Street	Yes	5	3	5
E	Blue Car with Reg. No. AB11_XYZ hit a Red Car in Oxford	Yes	7	5	3
F	A Silver Car with Reg. No. PQ11_RST hit an- other car on Cambridge Street	Yes	3	3	2
G	James’s car was involved in an accident on Cam- bridge St at 10:30 AM	Yes	5	8	4
H	James had an accident with his car in Cambridge at 10:30 AM	Yes	4	5	1
I	James has died in a road accident this morning at 10:30 AM	Yes	6	9	5
J	A Red Car has collided with a Silver car on Cam- bridge Street	Yes	4	7	3
K	Traffic accident on Cam- bridge Street with no ca- sualty	Yes	5	5	6
L	James Sent a Message at 5:00 PM	Yes	6	6	4

*Continued on next page*

Table B.1 – *Continued from previous page*

ID	Proposition	Satisfiable	Scores		
			Location	Freshness	Reputation
M	Traffic collision at Oxford Street, both drives are spot dead	Yes	1	5	1
N	Three people died after traffic collision at Oxford Street	Yes	2	4	5
O	Traffic collision at Oxford Street, 3 people seriously injured	Yes	4	2	3
P	Three people died in a traffic accident in Westminster	Yes	5	5	5
Q	Traffic accident in Westminster 1 person died	Yes	3	3	1
R	Car Drove into a bus stop shelter at Oxford Street. 3 people seriously injured	Yes	2	4	2
S	James was somewhere in Cambridge Street at 10 30am	Yes	4	3	4
T	James car hit another car at Cambridge Street at 10 30am	Yes	3	1	0
U	One person died and 2 got seriously injured in a road accident in Manchester	Yes	4	4	5
V	Traffic collision in Brighton : 3 casualties. Minimum 1 person died	Yes	4	5	3

*Continued on next page*



Table B.1 – *Continued from previous page*

ID	Proposition	Satisfiable	Scores		
			Location	Freshness	Reputation
W	Traffic accident in Banbury: minimum 1 dead or critically Injured	Yes	3	3	2
X	Major road accident in Coventry one person died and two were severely injured	Yes	5	8	4
Y	Traffic accident in Birmingham one person died and two were severely injured	Yes	4	5	1

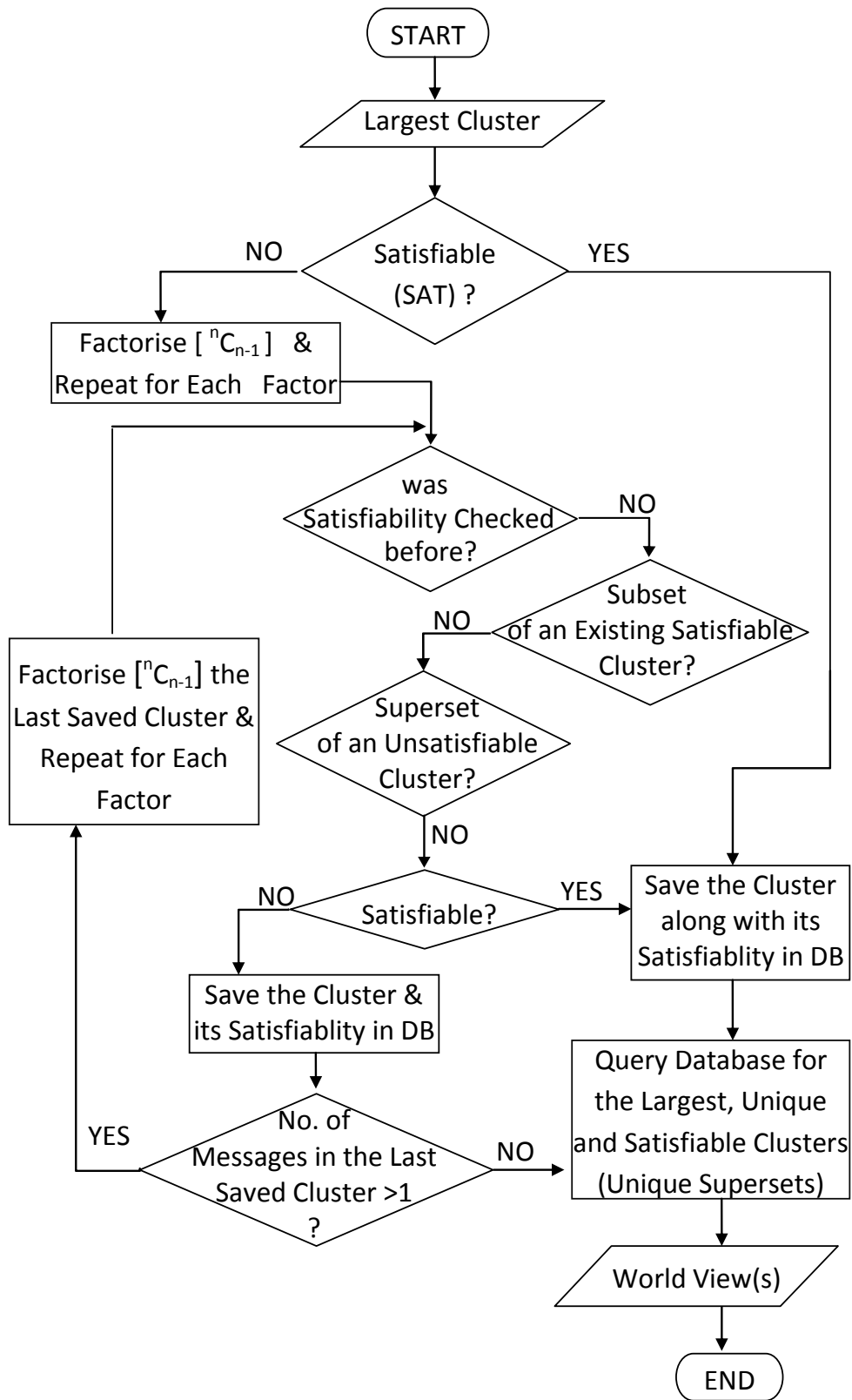


Figure B.1: Process Diagram of the World View Generator (Depth-First Search)

## B.2 World Views Generated in the Experiment in Chapter 9

The world views generated from the twenty messages listed in Table 9.4 and Table 9.7 are shown in Table B.2. The score of the world views and the messages included in each world view are also shown in the table.

Table B.2: World Views Generated from the Test Data Listed in Table 9.4 and Table 9.7

World View	Messages in World Views	Score
A F G I J S T	A Blue Nissan hit another Red Car	4.18
	F A Silver Car with Reg. No. PQ11_RST hit another car on Cambridge Street	
	G James's car was involved in an accident on Cambridge St at 10:30 AM	
	I James has died in a road accident this morning at 10:30 AM	
	J A Red Car has collided with a Silver car on Cambridge Street	
	S James was somewhere in Cambridge Street at 10 30am	
	T James car hit another car at Cambridge Street at 10 30am	
A F G J K S T	A Blue Nissan hit another Red Car	3.98
	F A Silver Car with Reg. No. PQ11_RST hit another car on Cambridge Street	
	G James's car was involved in an accident on Cambridge St at 10:30 AM	
	J A Red Car has collided with a Silver car on Cambridge Street	
	K Traffic accident on Cambridge Street with no casualty	
	S James was somewhere in Cambridge Street at 10 30am	
T James car hit another car at Cambridge Street at 10 30am		

*Continued on next page*

Table B.2 – *Continued from previous page*

<b>World View</b>	<b>Messages in World Views</b>	<b>Score</b>
A B D L N P	A Blue Nissan hit another Red Car	4.51
	B Car with Reg. No. AB11_XYZ hit a Red BMW on Oxford Street	
	D Blue Car with Reg. No. AB11_XYZ hit a Red Car on Oxford Street	
	L James Sent a Message at 5:00 PM	
	N Three people died after traffic collision at Oxford Street	
	P Three people died in a traffic accident in Westminster	
A B D L O R	A Blue Nissan hit another Red Car	4.01
	B Car with Reg. No. AB11_XYZ hit a Red BMW on Oxford Street	
	D Blue Car with Reg. No. AB11_XYZ hit a Red Car on Oxford Street	
	L James Sent a Message at 5:00 PM	
	O Traffic collision at Oxford Street, 3 people seriously injured	
	R Car Drove into a bus stop shelter at Oxford Street. 3 people seriously injured	
A B D I Q	A Blue Nissan hit another Red Car	4.39
	B Car with Reg. No. AB11_XYZ hit a Red BMW on Oxford Street	
	D Blue Car with Reg. No. AB11_XYZ hit a Red Car on Oxford Street	
	I James has died in a road accident this morning at 10:30 AM	
	Q Traffic accident in Westminster 1 person died	
A B D L M	A Blue Nissan hit another Red Car	4.19
	B Car with Reg. No. AB11_XYZ hit a Red BMW on Oxford Street	
	D Blue Car with Reg. No. AB11_XYZ hit a Red Car on Oxford Street	

*Continued on next page*

Table B.2 – *Continued from previous page*

<b>World View</b>	<b>Messages in World Views</b>	<b>Score</b>
	L James Sent a Message at 5:00 PM	
	M Traffic collision at Oxford Street, both drives are spot dead	
A B D L Q	A Blue Nissan hit another Red Car B Car with Reg. No. AB11_XYZ hit a Red BMW on Oxford Street D Blue Car with Reg. No. AB11_XYZ hit a Red Car on Oxford Street	4.12
	L James Sent a Message at 5:00 PM Q Traffic accident in Westminster 1 person died	
A F J K L	A Blue Nissan hit another Red Car F A Silver Car with Reg. No. PQ11_RST hit another car on Cambridge Street J A Red Car has collided with a Silver car on Cambridge Street K Traffic accident on Cambridge Street with no casualty	4.51
	L James Sent a Message at 5:00 PM	
A E I	A Blue Nissan hit another Red Car E Blue Car with Reg. No. AB11_XYZ hit a Red Car in Oxford I James has died in a road accident this morning at 10:30 AM	5.38
A E L	A Blue Nissan hit another Red Car E Blue Car with Reg. No. AB11_XYZ hit a Red Car in Oxford	4.93
	L James Sent a Message at 5:00 PM	
A H I	A Blue Nissan hit another Red Car H James had an accident with his car in Cambridge at 10:30 AM I James has died in a road accident this morning at 10:30 AM	4.81

### B.3 World Views Generated with a Smaller Scope

While Table 9.8 and Table B.2 show the world views generated within the scope of 3 Cars i.e. there are 3 cars available in the universe, Table B.3 shows the world views generated from the same data set but with only 2 cars available in the universe.

Table B.3: World Views Generated from the Test Data in Table 9.4 and Table 9.7 with only 2 Cars Available in the Universe (Scope)

<b>World View</b>	<b>No. of Messages in World View</b>
A B D L N P	6
A B D L O R	6
F G I J S T	6
F G J K S T	6
A B D I Q	5
A B D L M	5
A B D L Q	5
F J K L	4
A E I	3
A E L	3
A I S	3
A K L	3
A K S	3
H I	2

## APPENDIX C

# Alloy Representation of the Facts and Messages Used in Consistency Analysis

```
module RoadAccident
open util/ordering[CompositeTime]

sig Event
{
  loc: one Location, // it may have to be 'some' instead of 'one'.
  TSO defines this to be one or more i.e 'some'
  time: some CompositeTime,
  actor: some ActiveObject,
  casualty: set LivingObject
}{loc = actor.addr.time && (#casualty=0 || loc = casualty.addr.time
  )} //actor's location at the time of incident has to be the same
  as event location.

sig SendingMessage extends Event
{
  recipient: some Person,

}{#recipient>=1 && #actor=1 && (one actor & Person) && actor !=
  recipient && no casualty && (actor.action.time = SentMessage)}

sig TransportIncident extends Event{}
{ some actor & Vehicle
  loc in Road + RailTrack
```

```

one time    // Although a Transport Incident may last over a
            period of time but the 'time of incident' (i.e. the time when
            the
            // incident happened) refers to a specific moment in time.
            If a vehicle crashes and sets on fire and last over an
            hour,
            // yet we usually say it happened at 11am or 11:30 am.
}
sig Breakdown extends TransportIncident(){#actor = 1 }
sig Collision extends TransportIncident(){#actor = 2 && some
    Vehicle.driver} --Vehicle.driver >=1
sig Crash extends TransportIncident(){#actor = 2 }

sig Location {}    //Location CORRESPONDS to LOCTYPE in TSO

sig Road, RailTrack, Air, Waters, Jail, Hospital, Office, Home in
    Location{}

sig London, Oxford, Cambridge, Banbury, Coventry, Birmingham,
    Brighton, Manchester extends Location {}
sig BetnalGreen, Whitechapel, Westminster extends London{}
sig OxfordStreet extends Westminster{}
sig CambridgeStreet extends BetnalGreen{}

sig Object
{
    owner: set Person,
    addr:Location one -> set CompositeTime} //An object can be at a
        location in one or more moments in time but cannot be at
        two locations in the same moment in time
//Therefore, addr is a relation that is mapping one Location to
    zero or more moments in Time. BUT more than one location can
    never be mapped to the same moment in time.
sig PassiveObject extends Object {}
sig ActiveObject extends Object
{
    action: Action set -> set CompositeTime
}

sig BusStopShelter extends PassiveObject{}
sig LivingObject extends ActiveObject {healthCondition:
    HealthCondition one -> set CompositeTime}

```



```

//
*****

sig CompositeTime
{
    day: Int,
    hour: Int,
    minute: Int
}
{ //day > 0 && day <=31 && hour >=0 && hour <=23
    day > 0
    day <= 31
    hour >= 0
    hour <= 23
    minute >= 0
    minute <= 60
}
//
*****

abstract sig HealthCondition {time: lone CompositeTime}
one sig Well_N_alive extends HealthCondition {}
one sig SlightlyInjured extends HealthCondition {}
one sig SeverelyInjured extends HealthCondition {}
one sig Dead extends HealthCondition {}

abstract sig Colour {}
one sig Black, Blue, Green, Red, Silver extends Colour {}
abstract sig VehicleManufacturer {}
one sig BMW, Ford, Honda, Nissan, Toyota, Vauxhall, Volkswagen,
    Volvo extends VehicleManufacturer {}

abstract sig VehicleRegistration {}
one sig AB11_XYZ, AB22_XYZ, AB33_XYZ, PQ11_RST, PQ22_RST,
    PQ33_RST extends VehicleRegistration {}

sig Vehicle extends ActiveObject
{
    make: one VehicleManufacturer,

```

```

//model: Model
regNo: one VehicleRegistration ,
colour: one Colour, //suppose no car has more than one colour
      in our universe

driver: Person lone -> set CompositeTime, /*There can be only
      one driver at a given
time. Although the same driver can drive a vehicle at multiple
      moments
in time but multiple driver cannot drive at the same moment in
      time. At times,
there may be no driver in the vehicle at all. */
runsOn: some RailTrack+Road,
hit : Hit
} { addr.CompositeTime in runsOn}

sig Train, Tram, Bike, Car, Truck, Lorry extends Vehicle {}

sig Person extends LivingObject
{
      uses: set Object,
      home: set Home,
      wound: set Injury,
      locked_in: set Jail
}{no (uses & Person) && no owner}
sig Man, Woman, Boy, Girl extends Person {}
one sig James extends Man {}

sig Action {}
sig ConsciousAction extends Action {}
sig UnconsciousAction extends Action {}
sig ViolentAction extends Action{}
sig Walking, Running, Jumping, Talking, SentMessage extends
      ConsciousAction {}
sig GivingMessage extends ConsciousAction {recipient: Person}{#
      recipient >=1}
// Suppose, we're only concerned about the victims that are living
      being
sig Kick, Punch, Stab, Kill extends ViolentAction {victim:
      LivingObject}
sig Hit extends ViolentAction {victim: Object}
sig Injury {}

```

```

fact
{
all car: Vehicle | (car.regNo = AB11_XYZ) => (car.colour = Blue &&
    car.make = Nissan)
all car: Vehicle | (car.regNo = PQ11_RST) => (car.colour = Silver
    && car.make = Volkswagen) --&& car.owner= James)

all car: Vehicle | (car.regNo = PQ11_RST) <=> (#car.owner = 1) && (
    car.owner = James)
all car: Vehicle, t: CompositeTime | (car.regNo = PQ11_RST) <=> (
    car.driver.t = James)
all veh : Vehicle | #veh.owner = 1
all disj veh1, veh2: Vehicle| lone (veh1.owner & veh2.owner)
all veh1, veh2: Vehicle | (veh1.regNo = veh2.regNo) iff (veh1 =
    veh2) // Two vehicles cannot have same Registration No
all veh1, veh2: Vehicle | some (veh1.driver & veh2.driver) => (veh1
    = veh2) //Two vehicles cannot be driven by the same driver at
    the same time

//At a given time the location of a vehicle and its driver is same
all veh: Vehicle, t:CompositeTime |( some (veh.driver.t)) => veh.
    addr.t = (veh.driver.t).addr.t

//At the time of the accident, at least one of the vehicles
    involved in that accident had to have a driver.
all trafficIncident : TransportIncident | all t : trafficIncident.
    time | some veh : trafficIncident.actor & Vehicle | (one veh.
    driver.t)

all t1,t2: CompositeTime | lt[t1, t2] <=> (t1.day < t2.day) or ((
    t1.day = t2.day) && (t1.hour < t2.hour)) or ((t1.day = t2.day)
    && (t1.hour = t2.hour) && (t1.minute < t2.minute))
all t1,t2: CompositeTime | gt[t1, t2] <=> (t1.day > t2.day) or ((
    t1.day = t2.day) && (t1.hour > t2.hour)) or ((t1.day = t2.day)
    && (t1.hour = t2.hour) && (t1.minute > t2.minute))

// If a Living being dies now then it's dead ever afeter
all t1,t2: CompositeTime, l: LivingObject | (l.healthCondition.t1 =
    Dead) && lt[t1, t2] => (l.healthCondition.t2 = Dead)
// No Living object can take any action after death

```

```

all l: LivingObject, t: CompositeTime | (l.healthCondition.t = Dead
) => l.action.t not in ConsciousAction --l.behave
//If an actor does a concious action then he is not dead and he
wasn't dead before
all t1,t2: CompositeTime, l: LivingObject | ((l.action.t2 in
ConsciousAction) => (l.healthCondition.t2 != Dead)) && lt[t1, t2
] => (l.healthCondition.t1 != Dead)
}

/***/1***1***1***1***1***1***1***1***1***1***1***1***1***1*
//A Blue Nissan Car has hit another Red Car
pred Blue_Nissan_hit_another_Red_Car [roadAccident:Collision, car1,
car2: Vehicle]
{
car1 != car2
(car1 + car2) in roadAccident.actor
car1.make = Nissan
car1.colour = Blue
car2.colour = Red
}

/***/2***2***2***2***2***2***2***2***2***2***2***2***2**
// A Car with Reg# AB11_XYZ has hit another Red BMW on Oxford
Street
pred Car_AB11_XYZ_hit_Red_BMW_on_OxfordSt [roadAccident:Collision,
car1, car2: Vehicle]
{
car1 != car2
roadAccident.loc = OxfordStreet
(car1 + car2) in roadAccident.actor
car1.regNo = AB11_XYZ
--car1.colour = Blue
car2.colour = Red
car2.make = BMW

//If a message doesn't specify the time of incident then by
default, the time of incident
//will be the time when the message was received.
--roadAccident.time = Morning
}
run Car_AB11_XYZ_hit_Red_BMW_on_OxfordSt for 1 but 7 int, exactly 2
Car, exactly 2 Person

```











```

//
    ***13***13***13***13***13***13***13***13***13***13***13***13***13***13***13***13*

// Traffic collision at Oxford St: both drives are spot dead
pred Traffic_Collision_at_Oxford_St_Drivers_Spot_Death [roadAccident
    :Collision, car1, car2: Vehicle]
{
    car1 != car2
    roadAccident.loc = OxfordStreet
    (car1 + car2) in roadAccident.actor
    let t = roadAccident.time | (#roadAccident.casualty=2) && (
        roadAccident.casualty.healthCondition.t = Dead) && ((car1.
            driver.t + car2.driver.t) in roadAccident.casualty)
    && #Vehicle.driver =2
}
run Traffic_Collision_at_Oxford_St_Drivers_Spot_Death for 1 but 7
    int, exactly 2 Car, exactly 2 Person

//
    ***14***14***14***14***14***14***14***14***14***14***14***14***14***14***14***14*

// Traffic collision at Oxford St: 3 people died

pred Traffic_Collision_at_Oxford_St_3_peopl_died [roadAccident:
    Collision, car1, car2: Vehicle]
{
    car1 != car2
    roadAccident.loc = OxfordStreet
    (car1 + car2) in roadAccident.actor
    let t = roadAccident.time | (#roadAccident.casualty=3) && (
        roadAccident.casualty.healthCondition.t = Dead)
}
run Traffic_Collision_at_Oxford_St_3_peopl_died for 1 but 7 int,
    exactly 2 Car, exactly 3 Person

//
    ***15***15***15***15***15***15***15***15***15***15***15***15***15***15***15***15*

// Traffic collision at Oxford St: 3 people seriously injured

```











## APPENDIX D

# High Level Components of TSO Schema

Chapter 6 described the construction of TSO schema in Section 6.3. The high level elements of the complete TSO schema is shown here in Figure D.1 (as it has been referred to in Section 6.3.1). Since, in my Decision Support System, I have used the elements *<CONTEXT>* and *<EVENT>*, and some of their sub-elements only, the bottom two elements (*<RESOURCE>* and *<MISSION>*) of the TSO schema have not been expanded to show their child elements.

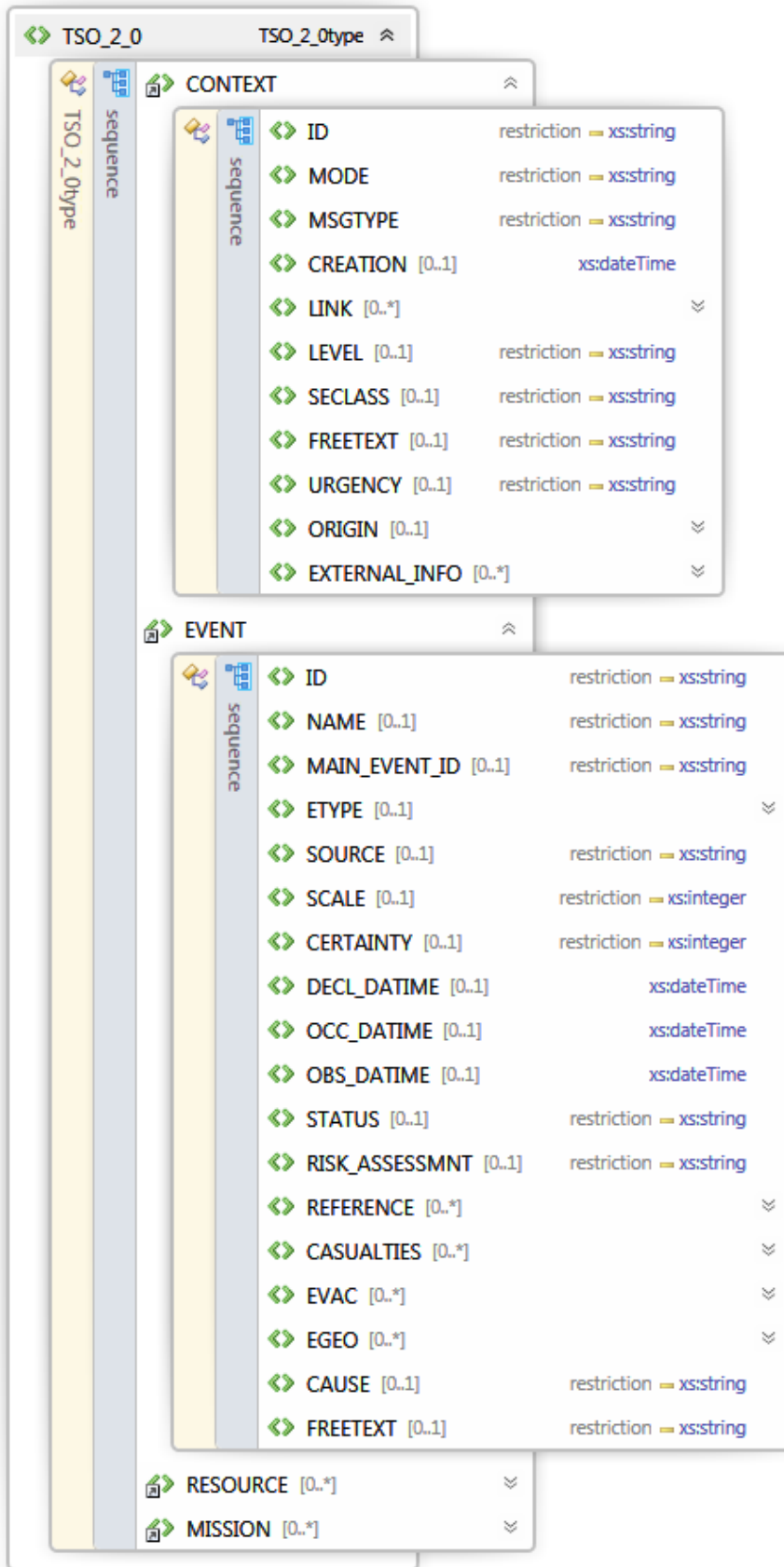


Figure D.1: High Level Components of TSO Schema  
231



## Functions and Stored Procedures used in the Back End

Fragments of the Decision Support System that are developed as Functions and Stored Procedures in the database are given in this chapter.

### E.1 Storing Tweets into Database

The following stored procedure is a part of the data harvesting mechanism of the Decision Support System (Unit 1 on Figure 7.1). It inserts each tweet into a database table. Twitter Search application that retrieves tweets based on the search criteria uses this stored procedure.

```

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
   This stored procedure inserts each tweet into a database table.
   Twitter Search application that retrieves tweets based on the
   search criteria uses this stored procedure.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * */
CREATE PROCEDURE [dbo].[spInsertStatusIntoTwitterSearch]
    @event_name NVARCHAR(50)
    ,@query_string NVARCHAR(300) = NULL
    ,@created_at DATETIME = NULL
    ,@tweet_id BIGINT = NULL
    ,@tweet_id_str NVARCHAR(30)
    ,@tweet_text NVARCHAR(150) = NULL
    ,@tweet_source NVARCHAR(250) = NULL
    ,@from_user NVARCHAR(50) = NULL
    ,@from_user_id BIGINT = NULL
    ,@from_user_id_str NVARCHAR(30)

```

```

,@geo NVARCHAR(200) = NULL
,@profile_image_url NVARCHAR(200) = NULL
,@to_user_id BIGINT = NULL
,@to_user_id_str NVARCHAR(30) = NULL
AS

IF EXISTS (SELECT 'True'
          FROM   TwitterSearch
          WHERE  (event_name = @event_name)
                AND (tweet_id_str = @tweet_id_str)
                AND (from_user_id_str = @from_user_id_str))
BEGIN
    --This means it exists, return it to ASP and tell us
    SELECT 'This record already exists!';
END
ELSE
BEGIN
    --This means the record doesn't exist, let's go ahead and
    add it
    SELECT 'Record Added';
    /* Create the new record */
    INSERT INTO TwitterSearch (
        event_name
    ,query_string
    ,created_at
    ,tweet_id
    ,tweet_id_str
    ,tweet_text
    ,tweet_source
    ,from_user
    ,from_user_id
    ,from_user_id_str
    ,geo
    ,profile_image_url
    ,to_user_id
    ,to_user_id_str
    )
VALUES (
    @event_name
    ,@query_string
    ,@created_at
    ,@tweet_id
    ,@tweet_id_str

```

```

    ,@tweet_text
    ,@tweet_source
    ,@from_user
    ,@from_user_id
    ,@from_user_id_str
    ,@geo
    ,@profile_image_url
    ,@to_user_id
    ,@to_user_id_str
)
END

```

## E.2 Scoring World Views

As mentioned in Table 8.1, the following stored procedure and functions implement the Combiner Function (Unit 7 on Figure 7.1) and score each world view.

```

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
   This stored procedure scores each of the world views.
* * * * * * * * * * * * * * * * * * * * * * * * * * * */

CREATE PROCEDURE [dbo].[sp_ScoreEachWorldView]
    @LocScoreMultiplier    REAL = 1,
    @FreshScoreMultiplier REAL = 1,
    @RepScoreMultiplier    REAL = 1
AS
    /*Reset all world views' previous scores to zero */
    UPDATE [MsgClusters] SET [Score] = 0

    /* Declare table-valued parameter for the stored procedure and
       then Call the stored procedure */
    DECLARE @WorldView CHAR_LIST_TBLTYPE

    DECLARE @WV_id  NVARCHAR(MAX),
            @WV_Score REAL

    /* The following WHILE Loop iterates through the entire talbe,
       read each world view and process it.
       However, the iteration process starts here: */

```

```

/* Read the world view having the Minimum (ASCII) value. The
   Min function is not vital here.
We could use the Max fuction too. The purpose of using the Min
   or (Max) funciton is simply
to provide a way that helps to iterate */

SELECT @WV_id = Min(MsgCluster)
FROM    MsgClusters
WHERE   worldview = 'true'

WHILE @WV_id IS NOT NULL
BEGIN
    /*We'll serialse a world view and then we will keep it in
       a Table_Valued_Parameter (i.e. @WorldView)
       But, remove all previous values from TVP before
       inserting elements of the new world view parameter*/
    DELETE FROM @WorldView

    /*Serialse the world view and keep it in a
       Table_Valued_Parameter (i.e. @WorldView) */
    INSERT INTO @WorldView (id) SELECT * FROM [Twitter].[dbo].[
        fn_SplitEachCharacter] (@WV_id)

    /* Invoke the function 'fn_GetScore4EachWorldView' to work
       out the score of the giiven world view (i.e. @WorldView)
       and assign the score to a varibale (@WV_Score) */
    SET @WV_Score = (SELECT [Twitter].[dbo].[
        fn_GetScore4EachWorldView] (@WorldView,
            @LocScoreMultiplier, @FreshScoreMultiplier,
            @RepScoreMultiplier))

    /*set the score of each world view */
    UPDATE [MsgClusters]
    SET [Score] = @WV_Score
    WHERE MsgClusters.MsgCluster = @WV_id

    /* Read the NEXT WORLD VIEW that has the next Minimum (
       ASCII) value */
    SELECT @WV_id = Min(MsgCluster)
    FROM    MsgClusters
    WHERE   MsgCluster > @WV_id AND worldview = 'true'
END

```

```

/*Show the output that was stroed in the table MsgClusters*/
SELECT * FROM MsgClusters
WHERE WorldView = 'true'
ORDER BY Score DESC

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
This (SQL Server) Function splits every character from a string.
In this application, each string is a world view ID and each
character is a message ID.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * */

CREATE FUNCTION [dbo].[fn_SplitEachCharacter](@String varchar(MAX))
returns @temptable TABLE (items varchar(1))
as
BEGIN
    DECLARE @slice varchar(1)
    if len(@String)<1 or @String is null return

        DECLARE @idx int = 1

    while len(@String) >= @idx
    BEGIN
        set @slice = substring(@String, @idx, 1)
        set @idx = @idx+1

        if(len(@slice)>0)
            insert into @temptable(items) values(@slice)
    END
    return
END

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
This (SQL Server) Function calculates and returns the score of
each world view
* * * * * * * * * * * * * * * * * * * * * * * * * * * * */

CREATE FUNCTION [dbo].[fn_GetScore4EachWorldView] (@TabValPara
CHAR_LIST_TBLTYPE readonly,
                                @LOC          INT,
                                @FRESH        INT,
                                @REP          INT)

```

```

RETURNS REAL
BEGIN
    RETURN
        (SELECT((SQRT(AVG(SQUARE(S.location_score))) * @LOC
            + Avg(CAST(S.freshness_score AS numeric(12,2))) * @FRESH
+ Avg(CAST(S.reputation_score AS numeric(12,2))) * @REP )/3)
        AS TotalScore
/* Casting of the value inside the AVG() function is necessary
because the AVG() aggregate function returns a value of the same
data type family as its argument. If casting was not done then
the AVG function would return an Integer discarding the decimal
places*/
        FROM    propositions P
            INNER JOIN tsoscores S
                ON P.tsoid = S.tsoid
        WHERE P.id IN (SELECT id
                        FROM    @TabValPara))
END

```

## APPENDIX F

# Alloy Automation and World View Generation Source Code

```
public class IncrementalGenerator : Generator
{
    #region [Variable Declarations]

    List<string> PropostionIdsNotInWViews = new List<string>();
    List<string> PropostionWithParametersNotInWViews = new List
        <string>();

    List<string> worldViewIds = new List<string>();
    List<string> propositionsInWorldViews = new List<string>();

    #endregion [Variable Declarations]

    public override void Process()
    {
        IEnumerable<Proposition> newPropositions = bc.
            GetAllNewSatisfiablePropositions();

        Console.WriteLine(String.Format("New Satisfiable
            Propositions: {0}", String.Join(",", newPropositions
                .Select(x => x.Id).ToArray())));

        //Read all old propostins
        //Alloy.WorldViewGenerator.GetSatisfiablePropositions()
        ;
        var allSatisfiablePropositions = bc.GetAllPropositions
            ().Where(x => x.IsSatisfiable);
    }
}
```

```

string temp = String.Join("",
    allSatisfiablePropositions.Select(x => x.Id));

//Append the new propositions with the old ones.
allPropositions.AddRange(newPropositions);
CreateDictionaryOfPropositions();

#region [World Views Generation Starts Here]

//foreach (string newPropId in newPropositionIds)
string newID;
string propositionIDs;
string newParameter;
string propositionsWithParameters;

// Lets create a list of AlloyInputParameters to send
across

foreach (var prop in newPropositions)
{
    newID = prop.Id;
    newParameter = prop.MessageWithParameters;

    //Get all Satisfiable Propostions that are not
    included in any World Views
bc.PropositionsNotInWViews(ref
    PropositionIdsNotInWViews, ref
    PropositionWithParametersNotInWViews);
//Retrieve all World Views and the Propostions
Included in each World View
bc.ReadWorldViews(ref worldViewIds, ref
    propositionsInWorldViews);

#region [World View Generation (Phase-1)]
//Combine the new message with all of the existing
propositions that are not part of any of the
existing
//world views and generate additional world views,
by feeding them into World View Generator.
if (PropositionIdsNotInWViews.Count > 0)
{

```



```

propositionIDs = newID + String.Join("",
    PropostionIdsNotInWViews);
//propositionsWithParameters.AddRange(
    oldPropostionWithParameters);
propositionsWithParameters = newParameter + "/"
    + String.Join("/",
    PropostionWithParametersNotInWViews);

//alloyParams.Add(new AlloyInputParams() {
    PropositionIds = propositionIDs,
    Propositions = propositionsWithParameters.
    Split('/') , NewId = newID });

// Create the input.als file by appending
    necessary statements to the Content of
    Template File
if (!bc.Contains(propositionIDs).HasValue)
{
    runner.CreateInputALSfile(
        propositionsWithParameters.Split('/'),
        inputFile);

    ///Remove the previous list of
        unsatisfiable propositions from 'strList
    ,
    ///before starting another round of
        checking and creating world views
    strList.Clear();

    // process the input file with Run Alloy
        Analyzer and Generate World Views
    ProcessDataAndGenerateWorldView(
        propositionIDs);
}
}

#endregion [World View Generation (Phase-1)]

#region [World View Generation (Phase-2)]
//Combine the new message with each of the existing
    world views to generate

```

```

//additional world views, by feeding them into
World View Generator.

for (int j = 0; j < worldViewIds.Count; j++)//
Wview in )
{
    propositionIDs = newID + worldViewIds[j];
    propositionsWithParameters = newParameter + "/"
        + propositionsInWorldViews[j];

    if (!bc.Contains(propositionIDs).HasValue)
    {

        // Create the input.als file by appending
        necessary statements to the Content of
        Template File
        runner.CreateInputALSfile(
            propositionsWithParameters.Split('/'),
            inputFile);

        ///Remove the previous list of
        unsatisfiable propositions from 'strList
        ,
        ///before starting another round of
        checking and creating world views
        strList.Clear();

        // process the input file with Run Alloy
        Analyzer and Generate World Views
        ProcessDataAndGenerateWorldView(
            propositionIDs);
    }
}
#endregion [World View Generation (Phase-2)]

#region [Move the processed proposition from
NewPropositions table to Propositions table]

bc.MoveFromNewToCurrentPropositions(newID);

#endregion [Move the processed proposition from
NewPropositions table to Propositions table]
}

```

```

        #endregion [World Views Generation Starts Here]
    }

}

public abstract class Generator : IGenerator
{
    protected AlloyRunner runner;
    protected GeneratorBC bc;

    protected string inputFile;
    protected string worldViewFile;

    public Generator()
    {
        bc = new GeneratorBC();

        string javaClassPath = ConfigurationManager.AppSettings["JavaClassPath"];
        string jdbcLibraryPath = ConfigurationManager.AppSettings["JDBCLibraryPath"];
        string package = ConfigurationManager.AppSettings["AlloyPackageName"];
        string templateFile = ConfigurationManager.AppSettings["AlloyTemplateFile"];

        this.inputFile = ConfigurationManager.AppSettings["InputAlsFile"];
        this.worldViewFile = ConfigurationManager.AppSettings["WorldViewFile"];

        runner = new AlloyRunner(javaClassPath, jdbcLibraryPath, package, templateFile);
    }
}

```

```

public List<Proposition> allPropositions = new List<
    Proposition>();
public Dictionary<string, Proposition>
    dictionaryPropositions = new Dictionary<string,
    Proposition>();

public void CreateDictionaryOfPropositions()
{
    dictionaryPropositions.Clear();

    foreach (var prop in allPropositions)
        dictionaryPropositions[prop.Id] = prop;
}

/// <summary>
/// Recent Change. May need to Undo last two actions
/// </summary>
public static List<string> strList = new List<string>();

public void ShowWorldViews()
{
    Console.WriteLine();
    Console.WriteLine("Showing World Views");
    Console.WriteLine();

    List<string> worldViews = bc.GetWorldViews();
    //Display the World Views on console window
    Console.WriteLine(string.Join("\n", worldViews.ToArray
        ()));

    //Sort the world views in order of their size and Save
    them in a Text file
    IEnumerable<string> sortedWorldViews =
        from element in worldViews
        orderby element.Length descending
        select element;

    int i = 0;
    string msgs = "";

    foreach (string view in sortedWorldViews)

```

```

    {
        i++;
        List<string> messagesInWorlView =
            AggregatePropositions(view);
        msgs += Environment.NewLine + " <--- WORLD VIEW:
            " + i.ToString() + " --->"
            + Environment.NewLine + Environment.NewLine
            + string.Join(Environment.NewLine,
                messagesInWorlView.ToArray())
            + Environment.NewLine + Environment.NewLine
            ;
    }

    StreamWriter sw = new StreamWriter(worldViewFile);
    sw.WriteLine(msgs);
    sw.Close();
}

public void Begin()
{
    var allSatisfiablePropositions = bc.GetAllPropositions
        ().Where(x => x.IsSatisfiable);
    this.allPropositions.AddRange(
        allSatisfiablePropositions);

    CreateDictionaryOfPropositions();

    string temp = String.Join("",
        allSatisfiablePropositions.Select(x => x.Id));

    Process();
}

public abstract void Process();

public void ProcessDataAndGenerateWorldView(string
    propositionIDs)
{
    try
    {
        //Console.WriteLine("Application Running .....");
    }
}

```

```

    ///3. Run AlloyAnalyzer to evaluate the composite
        predicate
    ///

    bool? alloyOutput = runner.RunService(inputFile);

    if (!alloyOutput.HasValue)
    {
        Console.WriteLine("Error.....");
    }
    else
    {
        ///
        ///4. Save the result received from Alloy
            regardless whether the composite predicate
            is satisfiable or not.
        //TODO: Remove comment below
        bc.Insert(propositionIDs, alloyOutput.Value);
        Console.WriteLine(String.Format("Proposition:
            {0}, Satisfiable: {1}", propositionIDs,
            alloyOutput.Value));
        ///If it was satisfiable then we have found a
            world view of the largest possible size in
            step 4
        /// if (alloyOutput == "true") then '
            propositionIDs' is a world view
        ///
        ///5. If it was not satisfiable then call the
            method Factorise with the unsatisfiable
            cluster as a parameter (for example, ABCDEF)
            .
        if (!(alloyOutput.Value))
        {
            strList.Add(propositionIDs);
            for (int indx = 0; indx < strList.Count;
                indx++)
                Factorise(strList[indx]);
        }
        ///6. (Step 6 is inside the method 'Factorise')
    }
}
catch (Exception ex)
{

```

```
}  
}
```

```
public void Factorise(string inputStr)  
{  
    Console.WriteLine(inputStr);  
    //If inputStr (i.e. the all propositions together) doesn  
    't satisfy then  
    string s;  
    bool subSetOfSatisfiableCluster = false;  
    bool superSetOfUnsatisfiableCluster = false;  
    //List<string> strList = new List<string>();  
    for (int i = 1; i <= inputStr.Length; i++)  
    {  
        s = inputStr.Remove(inputStr.Length - i, 1);  
  
        //strList.Add(inputStr.Remove(inputStr.Length - i,  
        1));  
        if (s.Length >= 2)  
        {  
            /// Check if this cluster was checked ever  
            before  
            /// OR if the message cluster in hand is a  
            Subset of an Existing Satisfiable Cluster  
            /// OR if it is a Superset of an Unsatisfiable  
            Cluster?  
            if (!bc.Contains(s).HasValue) //i.e. if s wasn'  
            t found in the database (which means it wasn'  
            't checked before)  
            {  
                //Check if the message cluster in hand is a  
                Subset of an Existing Satisfiable  
                Cluster  
                var clusters = bc.GetAllMsgClusters();  
  
                IEnumerable<string> clustersForSubset =  
                clusters.Where(x => x.IsSatisfiable && x  
                .Cluster.Length >= s.Length).Select(x =>  
                x.Cluster);
```

```

subSetOfSatisfiableCluster = bc.IsSubSet(s,
    clustersForSubset);

if (!subSetOfSatisfiableCluster)
{
    IEnumerable<string> clustersForSuperSet
        = clusters.Where(x => !x.
            IsSatisfiable && x.Cluster.Length <=
                s.Length).Select(x => x.Cluster);
    //Check if the message cluster in hand
    is a Superset of an Unsatisfiable
    Cluster
    superSetOfUnsatisfiableCluster = bc.
        IsSuperSet(s, clustersForSuperSet);

    if (!superSetOfUnsatisfiableCluster)
    {
        /// 6. Send this sub-Cluster, s, to
        Alloy Analyzer.
        /// But sending a sub-Cluster to
        Alloy Analyzer means recreating
        the input.als with the sub-
        Cluster
        ///
        List<string> propositionCluster =
            AggregatePropositions(s);
        runner.CreateInputALSfile(
            propositionCluster.ToArray(),
            inputFile);

        //Run Alloy on the sub-cluster
        created in step 6
        bool? alloyOutput = runner.
            RunService(inputFile);

        #region Process Alloy Output
        if (!alloyOutput.HasValue)
        {
            Console.WriteLine("Error
                .....");
        }
        else
        {

```



```

    /// 7. Store the sub-Cluster, s
    , along with the returned
    value from Alloy into the
    Database
bc.Insert(s, alloyOutput.Value)
    ;
Console.WriteLine(String.Format
    ("Proposition: {0},
    Satisfiable: {1}", s,
    alloyOutput.Value));

    ///If it was satisfiable then
    we have found a world view
    of the largest possible size
    as in step 4
    /// i.e. if (alloyOutput == "
    true") then 's' is a world
    view
    ///
    /// 8. If s was not satisfiable
    and its length > 2 then,
    Add it to the strList [i.e.
    strList.Add(s);] Otherwise,
    nothing needs to be done
    /// if there are only two
    messages in s i.e. its
    length==2, then each of its
    messages will individually
    form a world view.
    /// It's unnecessary to
    check the satisfiability of
    each individual message again
    because the simple
    predicates that we used to
    /// form compound
    predicates are all
    satisfiable (their
    satisfiability was checked
    in the beginning of the
    process)
if (alloyOutput == false && s.
    Length > 2)

```

```

        strList.Add(s);
    }
    #endregion
}
}
}
}
}

public List<string> AggregatePropositions(string ids)
{
    List<string> propositonList = new List<string>();
    foreach (char ch in ids)
    {
        var prop = this.dictionaryPropositions[ch.ToString
            ()];
        propositonList.Add(prop.MessageWithParameters);
    }
    return propositonList;
}
}
}

```