Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

# Multiple Hypothesis Situation Analysis

## *Governing Concepts and Support System Prototype*

*Alexandre Bergeron Guyard*
*Jean Roy*
*DRDC Valcartier*

**Defence R&D Canada – Valcartier**
Technical Report
DRDC Valcartier TR 2010-525
February 2011

Canada

# Multiple Hypothesis Situation Analysis

*Governing Concepts and Support System Prototype*

Alexandre Bergeron Guyard
Jean Roy
DRDC Valcartier

## Defence R&D Canada – Valcartier

Technical Report
DRDC Valcartier TR 2010-525
February 2011

Principal Author

Alexandre Bergeron Guyard
Defence Scientist

Approved by

Stéphane Paradis
Head/Section Intelligence and Information

Approved for release by

Christian Carrier
Chief Scientist

## Abstract

In 2008, DRDC undertook Applied Research Project (ARP) 11hk entitled "Multiple Hypothesis Link Analysis for Anomaly Detection in the Maritime Domain" (MHLA-4-ADMD). The main objective of this ARP is to study, develop and implement a situation analysis tool to support anomaly detection, the identification of vessels of interest (VOI), and threat analysis in the maritime domain. In this context, it was decided to explore the concept of Multiple Hypothesis Situation Analysis (MHSA) and to develop a MHSA Support System (MHSASS) prototype. This document is meant to provide a complete overview of MHSA. The objective is twofold. Firstly, provide a theoretical look at MHSA, and discuss its governing principles. Secondly, provide detailed information on the MHSASS prototype that was developed. This document explains the concepts driving MHSA. It shows how MHSA works and why it could be of great use to the operators. This work constitutes a milestone that could lead to the creation of a system that would help analysts deal with the uncertainty that makes (even simple) situation difficult to analyse.

## Résumé

En 2008, RDDC a entrepris un projet de recherche appliqué (PRA) intitulé "*Multiple Hypothesis Link Analysis for Anomaly Detection in the Maritime Domain*" (MHLA-4-ADMD). L'objectif principal de ce PRA est d'étudier, de développer et d'implémenter un outil d'analyse de la situation pour supporter la détection d'anomalies, l'identification de *vessels of interest* (VOI) et l'analyse de la menace dans le domaine maritime. Dans ce contexte, il fut décidé d'explorer le concept de *Multiple Hypothesis Situation Analysis* (MHSA) et de développer un prototype de MHSA *Support System* (MHSASS). Ce document a pour but de fournir un survol complet de MHSA. Un objectif est de donner un aperçu théorique de MHSA, et de ses principe gouvernants. Un second objectif est de fournir de l'information détaillée sur le prototype MHSASS qui a été développé. Ce document explique les concepts qui guident le MHSA. Il montre comment le MHSA fonctionne et pourquoi il pourrait être d'une grande utilité pour les opérateurs. Ce travail constitue un point tournant qui pourrait mener à la création d'un système qui pourrait aide les analystes à gérer l'incertitude qui rend les situations (même les plus simples) difficiles à analyser.

This page intentionally left blank.

# Executive summary

## Multiple Hypothesis Situation Analysis

**Bergeron Guyard, A.; Roy, J.; DRDC Valcartier TR 2010-525; Defence R&D Canada – Valcartier; February 2011.**

**Introduction:** In 2008, DRDC undertook Applied Research Project (ARP) 11hk entitled "Multiple Hypothesis Link Analysis for Anomaly Detection in the Maritime Domain" (MHLA-4-ADMD). The main objective of this ARP is to study, develop and implement a situation analysis tool to support anomaly detection, the identification of vessels of interest (VOI), and threat analysis in the maritime domain. In this context, it was decided to explore the concept of Multiple Hypothesis Situation Analysis (MHSA) and to develop a MHSA Support System (MHSASS) prototype.

This document is meant to provide a complete overview of MHSA. It provides a theoretical look at MHSA, and discusses its governing principles. It also provides detailed information on the MHSASS prototype that was developed.

**Results:** This document explains the concepts driving MHSA. It shows how MHSA works and why it could be of great use to the operators. The illustrated results show how the MHSASS prototype was built, and how it reaches two distinct objectives: 1) it demonstrates the concept and power of MHSA when eventually put in the hands of the operators: the added value of helping the analysts handle multiple hypotheses about an uncertain situation, allowing to defer decisions until new evidence comes in and confirms certain hypotheses; 2) it also acts as a teaching tool, to help explain the various, sometimes complex, aspects of MHSA.

A survey of link analysis tools and an integration study are also discussed.

**Significance:** The developed MHSASS prototype fulfills its objectives. It provides a means to showcase the value of MHSA for the operators. It also provides a basis on which a full fledged MHSASS could be built. Indeed, although the current MHSASS is a prototype, its multiple hypothesis engine is reusable. Moreover, a study detailing how it could be integrated into an existing Link Analysis tool has been conducted. This work constitutes a milestone that could lead to the creation of system that would help analysts deal with the uncertainty that makes (even simple) situation difficult to analyse.

**Future plans:** There are plans to integrate the MHSA functionality into the Multi-Intelligence Tool Suite (MITS) developed by DRDC Valcartier. The MITS is a collection of integrated intelligence analysis tools. A MHSASS integrated into the MITS could leverage from automated reasoning and other Intelligence analysis functionalities, while providing a MHSA service.

# Sommaire

## Multiple Hypothesis Situation Analysis

**Bergeron Guyard, A.; Roy, J.; DRDC Valcartier TR 2010-525; R & D pour la défense Canada – Valcartier; Février 2011.**

**Introduction ou contexte:** En 2008, RDDC a entrepri un projet de recherche appliqué (PRA) intitulé "*Multiple Hypothesis Link Analysis for Anomaly Detection in the Maritime Domain*" (MHLA-4-ADMD). L'objectif principal de ce PRA est d'étudier, de développer et d'implémenter un outil d'analyse de la situation pour supporter la détection d'anomalies, l'identification de *vessels of interest* (VOI) et l'analyse de la menace dans le domaine maritime. Dans ce contexte, il fut décidé d'explorer le concept de *Multiple Hypothesis Situation Analysis* (MHSA) et de développer un prototype de MHSA *Support System* (MHSASS).

Ce document a pour but de fournir un survol complet de MHSA. Un objectif est de donner un aperçu théorique de MHSA, et de ses principe gouvernants. Un second objectif est de fournir de l'information détaillée sur le prototype MHSASS qui a été développé.

**Résultats:** Ce document explique les concepts qui guident le MHSA. Il montre comment MHSA fonctionne et pourquoi il pourrait être d'une grande utilité pour les opérateurs. Les résultats illustrés montrent comment le prototype MHSASS a été construit, et comment il atteint deux objectifs distincts: 1) démontrer le concept et la puissance de MHSA lorsque mis entre les mains d'un opérateur, la valeur ajoutée de supporter les analystes dans la manipulation d'hypothèses multiples au sujet d'une situation incertaine, leur permettant de repousser les décisions jusqu'à ce que de nouvelles informations confirment certaines hypothèses; 2) agir comme un support pédagogique, pour permettre d'illustrer les différents aspects, parfois complexes, de MHSA.

Une revue des outils d'analyse de liens et un étude d'intégration sont aussi discutées.

**Importance:** Le prototype MHSASS rencontre ses objectifs. Il permet de démontrer la valeur de MHSA pour les opérateurs. Il donne aussi une base sur laquelle une système MHSASS complet pourrait être construit. Bien que le MHSASS actuel ne soit qu'un prototype, son moteur de gestion d'hypothèses multiples est réutilisable. De plus, une étude d'intégration dans un outil d'analyse de liens a été effectuée. Ce travail constitue un point tournant qui pourrait mener à la création d'un système qui pourrait aider les analystes à gérer l'incertitude qui rend les situations (même les plus simples) difficiles à analyser.

**Perspectives:** L'intégration d'une fonctionnalité MHSA dans le *Multi-Intelligence Tool Suite* (MITS), développé par RDDC Valcartier est planifiée. Le MITS est une collection d'outils intégrés d'analyse du renseignement. Un outil MHSASS intégré au MITS pourrait bénéficier des fonctionnalités de raisonnement automatique, ainsi que d'autres fonctionnalités, tout en offrant un service de MHSA.

# Table of contents

# List of figures

# List of tables

# 1 Introduction

In 2008, DRDC undertook Applied Research Project (ARP) 11hk entitled “Multiple Hypothesis Link Analysis for Anomaly Detection in the Maritime Domain” (MHLA-4-ADMD). The main objective of this ARP is to study, develop and implement a situation analysis tool to support anomaly detection, the identification of vessels of interest (VOI), and threat analysis in the maritime domain. In this context, it was decided to explore the concept of Multiple Hypothesis Situation Analysis (MHSS) and to develop a MHSA Support System (MHSASS) prototype.

This document is meant to provide a complete overview of MHSA. This objective is twofold. Firstly, provide some theoretical look at MHSA, and discuss its governing principles. Secondly, provide detailed information on the MHSASS prototype that was developed.

Section 2 provides a quick discussion on Situation Analysis, how it can be challenging, and how MHSA can support this process. Section 3 explains MHSA governing concepts. It starts by introducing a number of notions required to understand MHSA. It then moves on to detail the Situation Description Language (SDL) and Multiple Hypothesis Tree (MHTree) used to depict a situation and derive and maintain hypotheses. Section 3.4 provides detailed information on the SDL and its intricacies, especially the specification and management of the dependencies between the situation model components, which is a crucial but complex aspect of the MHSA approach. Section 4 gives a detailed overview of the development work that was done for the MHSASS prototype. Among other things, prototype requirements, design and configuration are discussed. Section 5 discusses a survey of link analysis tools and the integration study that were also conducted.

# 2 Situation Analysis

Situation analysis is defined as a process dealing with the examination of a situation, its elements, and their relations, to provide and maintain a state of situation awareness for the decision maker(s) [1]. The situation analysis process is concerned with understanding the world. There is a real situation in the environment and the situation analysis process aims at creating and maintaining a mental representation of it for the operator.

At the highest level, the situation analysis process can be decomposed into four sub-processes [2]:

1) situation perception;
2) situation comprehension;
3) situation projection; and
4) situation monitoring.

Situation perception has to do with the "acquisition" of the situation through data and information collection with various sensors and other sources. Situation comprehension is about further developing one's knowledge of the situation with respect to both its nature (i.e., the inherent character or basic constitution of the situation) and its significance or meaning (i.e., the importance of the situation). This sub-process must be able to grasp the nature of the situation and to derive operationally relevant meaning and significance from the results of situation perception. Situation projection must produce estimates of future possibilities for situation elements, based on current trends, and of the consequences, impact, or the implications of the situation. Finally, situation monitoring has to do with watching, observing, or checking the evolution of the situation in order to keep track of, regulate, or control the operation of the situation analysis process.

A huge portion of the efforts deployed by the situation analysis practitioners in the information fusion domain involves dealing with the unavoidable uncertainty associated with the knowledge, information, and/or data provided by the different sources. In the presence of uncertainty, the analysis of even simple situations can quickly become very complex. Facing this uncertainty, the situation analyst will necessarily have to formulate multiple hypotheses regarding the situation, each hypothesis expressing a unique resolution of all uncertain variables [3]. Because of cognitive limitations, he/she can quickly lose track of all of the possibilities. There is thus a need to organize the hypotheses such that it becomes easier to visualize and manage them.

The MHSASS addresses the cognitive overload problem by helping the operators maintain multiple representations of possible situations.

# 3 Multiple Hypothesis Situation Analysis Governing Concepts

This section addresses the general concepts surrounding Multiple Hypothesis Situation Analysis (MHSA). In order to help operators maintain multiple representations of possible situations it is first necessary to provide them with a means to explicitly and formally describe their mental model of a specific situation. A situation can be defined as a specific combination of circumstances, i.e., conditions, facts, or states of affairs, at a given moment. This representation must include the various elements of the situation, the relationships and dependencies present between these elements, and the potential uncertainty relative to any element or relationships. It is therefore required to provide the analysts with a language appropriate to express such a representation. Link analysis (LA) tools handle similar representation needs. Figure 1 shows a display example of the Collation and Link Analysis (CoALA) tool.Various elements of a situation are displayed, along with relations between these various elements. However, LA tools sometimes offer limited functionality to express uncertainty, and possible different situations (cf. 5.1).

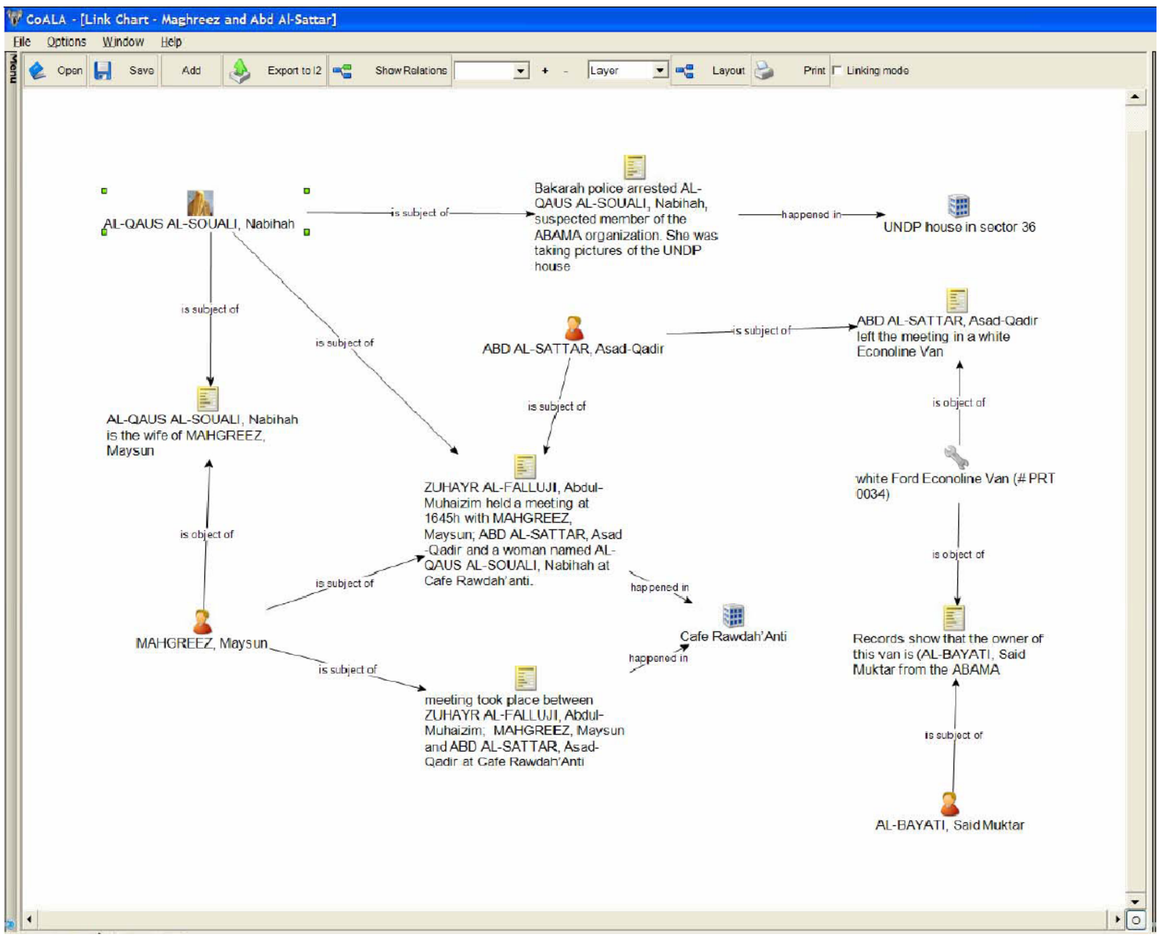



*Figure 1 CoALA: Collation and Link Analysis*

Starting from a formal representation of a situation and its uncertainty, it is possible to derive all the possible "real-life" scenarios from the analyst's initial mental model. This could be achieved with the use of a multiple hypothesis tree. The tree manages the various possible representations, and outputs the various hypotheses. This process, of transforming a situation model into a hypothesis tree to generate possible hypotheses, is explained in detail throughout this chapter, and is also documented in [3] [4].

## 3.1 Situation Model

As just mentioned, a situation is a combination of situation components at a particular moment. Figure 2 shows an example of a simple situation.

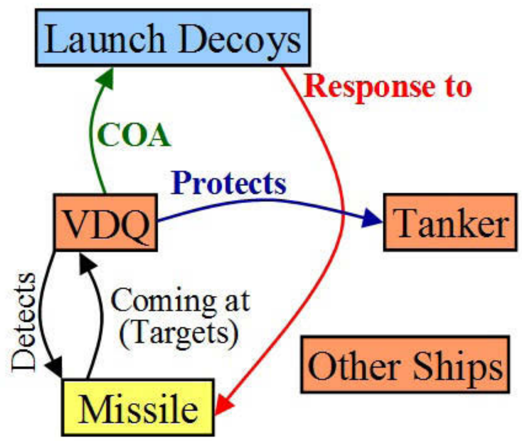



*Figure 2 Simple situation*

In this particular example, we are looking at a frigate, the Ville de Québec (VDQ on the figure) which is protecting a tanker. The VDQ detects a missile coming at her. In response to the missile, the VDQ launches decoys as its course of action (COA).

The situation depicted in Figure 2 is rather simplistic. Everything about the situation is known with absolute certainty. There is only one potential mental model of this situation and an operator could easily manage such a mental representation. Figure 3 shows a more complex example.

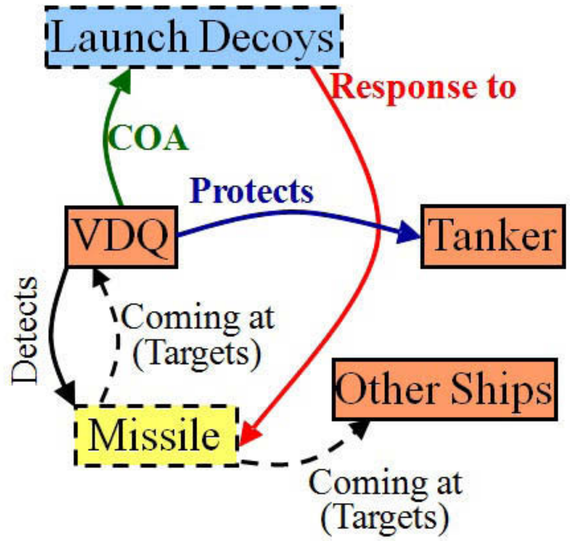



*Figure 3 More complex situation*

The dotted lines on the figure reflect uncertainty, i.e., it is unsure whether the elements represented with a dotted line are present in the real world or not. Once again, we have the VDQ which is protecting a tanker (this is known with certainty). The VDQ detects what could be a missile or a false alarm. If a missile is indeed present, it is unknown whether it is targeting the VDQ, the tanker or some other ship. The VDQ may turn around, launch decoys or use hard kill weapons if there is a missile targeting the tanker or the VDQ. However, if there isn't a missile, or if the missile is targeting some other ship, the VDQ will do nothing.

The situation represented in Figure 3 is more complex. There is uncertainty about many components of the situation. Therefore, many hypothetical situation models could be derived from this situation (Figure 4).

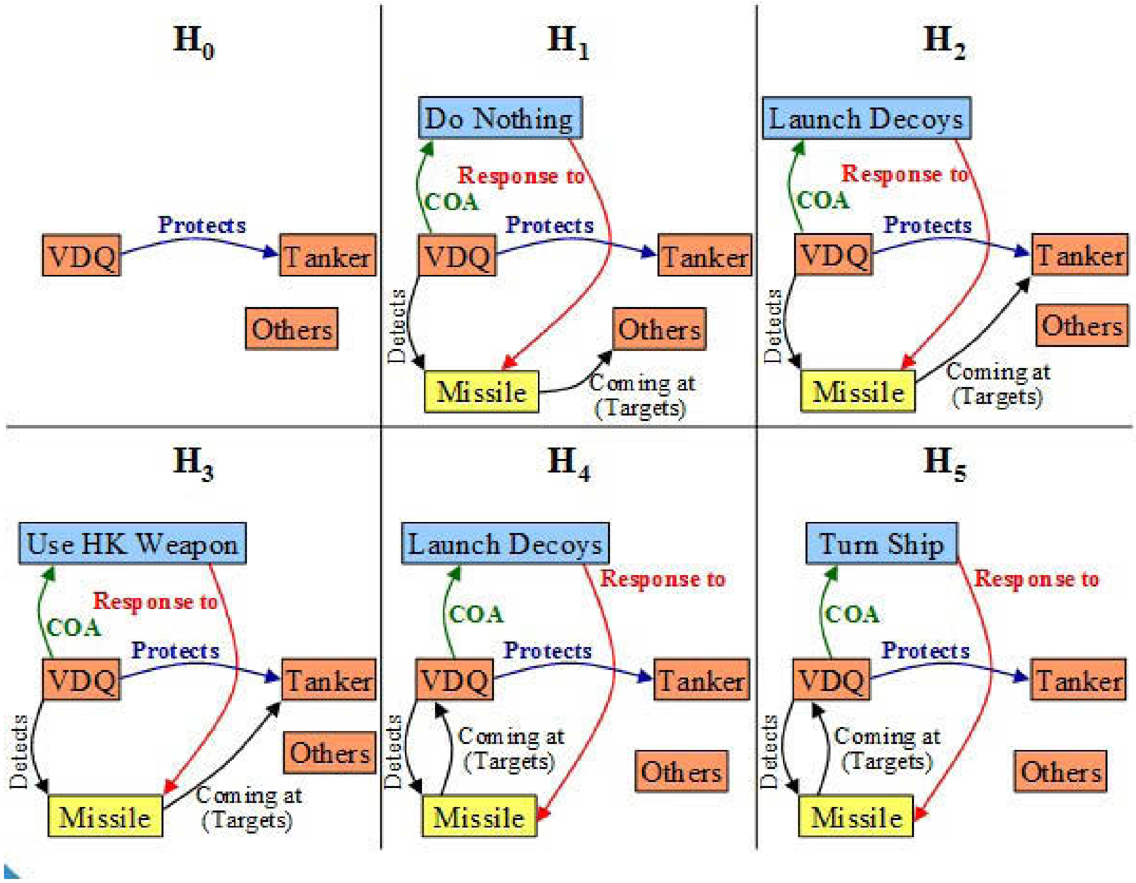



*Figure 4 Multiple hypothetical situations*

There are six potential real-world situations (or situation models) that could be derived from the uncertain situation depicted on Figure 3. Numbered from $H_0$ to $H_5$, the derived hypotheses could be described as follows:

The VDQ is protecting the tanker and...

$H_0$: no missile is present (i.e., the contact is a false alarm);

$H_1$: a missile is present and is targeting some other ship; the VDQ does nothing;

$H_2$: a missile is present and is targeting the tanker; the VDQ launches decoys;

$H_3$: a missile is present and is targeting the tanker; the VDQ uses hard kill weapons;

$H_4$: a missile is present and is targeting the VDQ; the VDQ launches decoys;

$H_5$: a missile is present and is targeting the VDQ; the VDQ turns around.

There are six potential real-world models for this situation and an operator may have to struggle to maintain them mentally. It is easy to imagine an even more complex situation with more components and uncertainty. As situations grow more complex, the number of potential real-world situations that can be derived grow exponentially and it quickly becomes impossible for a human to maintain a mental representation for each of them.

## 3.2 Situation Description Language

To help analysts create and maintain multiple situation models, it is important to provide them with a language to represent a situation and its uncertainty. Such a language, different from the one used for Bayesian Networks [5], was developed for the MHSASS. The following is a description of the graphical situation description language (SDL). Specific details as to how one can use the SDL in the MHSASS application are given in subsections 4.3.2 and 4.3.3.

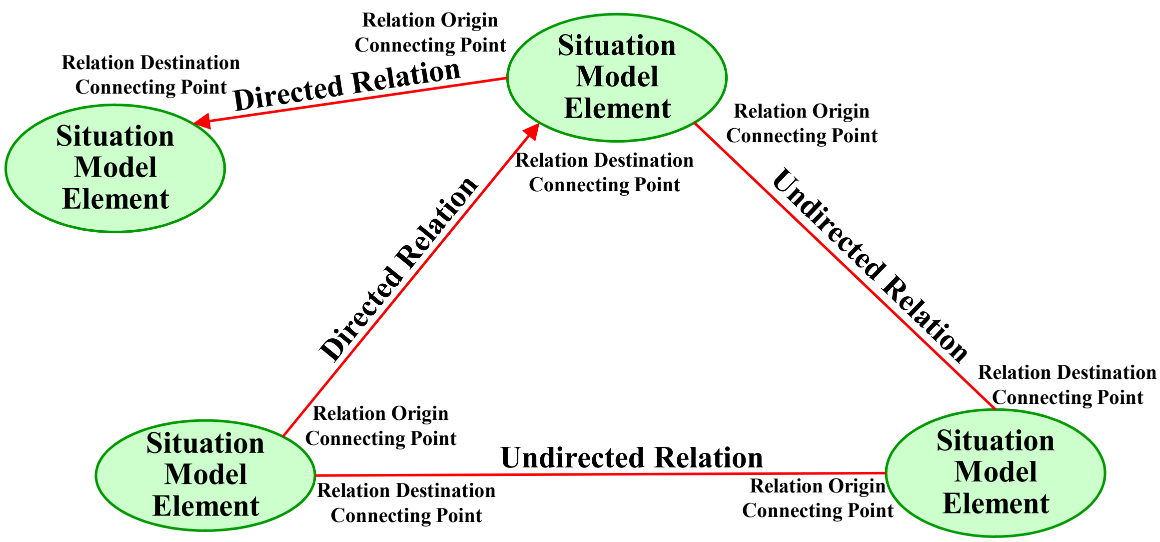



*Figure 5 Situation model description*

For the MHSASS, a situation is described using a situation model (Figure 5), which situation model is composed of situation model components (SMC). These can be one of the following:

Element

Undirected Relation

Directed Relation

Relation origin connecting point

Relation destination connecting point

The basic constructs of the language allow the analyst to specify the elements present in a situation. Once at least two elements have been defined, it is possible for the analyst to specify relations between these elements. A relation can only be present between two elements. One of these elements must be specified as the relation origin connecting point, the other as the relation destination connecting point. Relations can be directed or undirected.

Elements and relations (only) are place holders (or containers). As such, they don’t by themselves convey any particular semantic related to the situation being modeled. It is the actual contents of the situation model components that make sense (or not) to a human analysts. The MHSA support system manipulates the place holders (or containers), not the contents. Hence the system doesn’t care about the semantic of the contents. For the system, these contents are totally irrelevant. It is totally up to the system user to use the SDL to properly reflect a situation that "makes sense" and reflects their mental model. As the system will handle situation model components (and not the semantics), the user has to ensure proper use of the SDL in order to retrive valuable information from the system.

### 3.2.1 Situation Model Components' Uncertainty

The SDL also allows the analysts to express uncertainty about situation components. There is uncertainty (in the sense of MHSA) when there is more than one possibility (2, 3, 4,...) for a given situation model component. On the contrary, when there is only one possibility for a situation model component, then this possibility is considered as certain. It should be noted that when multiple possibilities are present for a component,  they must be mutually exclusive, i.e., if one is true, the others are false.

The SDL support three types of uncertainty:

existence uncertainty,

content uncertainty,

existence and content uncertainty.

Existence uncertainty refers to the idea that an element or relation of the situation model may or may not exist in the real world. Referring to the example of Figure 3, the missile has existence uncertainty, as it is either present or not in the real world. As previously mentioned, the origin and destination of a relation are mandatory in order for this relation to exist; as such, they cannot have existence uncertainty.

Content uncertainty refers to the notion that an element or relation of the situation model may have different meanings, or labels attached to it. Referring to the example of Figure 3, the box containing the VDQ's COA has content uncertainty as it either contains "do nothing", "use HK weapon", "turn around", or "launch decoys".

Existence and content uncertainty is a combination of the first two kinds of uncertainty. Actually, the box (of Figure 3) containing the VDQ's COA is an example of this third kind of uncertainty as it reflects content uncertainty (H1 to H5 in Figure 4), and it could also not exist (H0 in Figure 4).

As a practical convention, when the uncertainty type is “existence” or “existence & content”, then the first "content possibility” will always be “not existent”.

### 3.2.2 Dependency Requirements for Situation Model Components' Possibilities

The SDL allows the analyst to specify dependency requirements for the various possibilities of a SMC. Let's look at an example from Figure 4. Looking at the initial scenario, we know that the VDQ is to do nothing if the missile is targeting some other ship. So, in order for the COA element to have the content "do nothing", the missile has to be targeting some other ship. In other words, the possibility "do nothing" of the COA element ***requires*** the missile to be targeting "other ship". This is an example of SMC possibilities dependency requirements.

The SDL's possibility dependency requirements allows the user to specify the impact that the various possibilities of the different situation model components have on one another. It is possible to specify whether a SMC possibility ***requires*** another (or many others) SMC possibility to be true or false. Correspondingly, it is possible to specify if a SMC possibility causes another (or many others) SMC possibility to be true or false (i.e., "affects" it). This crucial aspect of the SDL provides greater expressivity to the user. However this expressivity comes at a price. Possibilities dependency requirements interact with on one another in a way that requires a careful management of SMCs. This is the topic of subsection 3.4, where the various types of SMC possibility dependency requirement interactions are discussed.

### 3.2.3 Situation Model Components' Possibilities Likelihood

When specifying SMC possibilities, the analyst can also specify a level of likelihood for every possibility. This measure of likelihood is used to compute the likelihood of global hypotheses. For the current version of the MHSASS prototype, a probability score is used to reflect the likelihood of a possibility. However, the prototype is designed in such a way that alternative mechanisms could be used to measure uncertainty.

## 3.3 Multiple Hypothesis Tree

Because of cognitive limitations, the analyst can quickly lose track of all of the possibilities of a given situation. There is thus a need to organize the hypotheses in such a way that it becomes easier to visualize and manage them. The SDL provides a way for the user to describe a situation and its uncertainty. A multiple Hypothesis Tree (MHTree) data structure can be used to organize the various hypotheses that can be derived from the described situation. As discussed in section 4.2, this tree structure is inspired by Multiple Hypothesis Tracking approaches ([3][6][7]).

### 3.3.1 Tree Structure

A tree is a data structure that emulates a hierarchical structure with a set of linked nodes. Mathematically, a tree is an acyclic connected graph where each node has zero or more children nodes and at most one parent node. Figure 6 shows examples of trees.

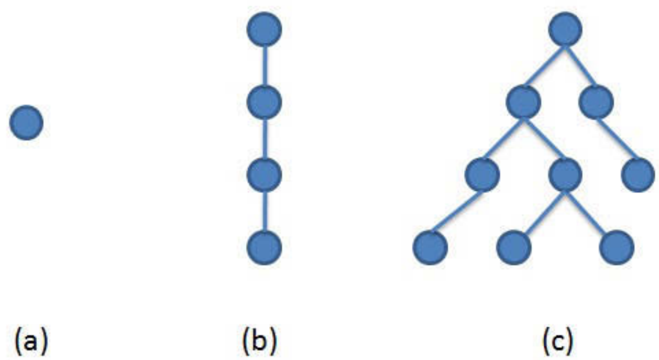



*Figure 6 Examples of tree data structures*

Example (a) represents a single node tree, without any children nodes. Example (b) depicts a tree with a single child for every parent node. This kind of simplified tree can also be referred to as a chain. Finally, example (c) shows a tree composed of a root node, 4 other parent nodes, and 4 "childless" or leaf nodes. Figure 7 illustrates a hypothesis tree, corresponding to the hypotheses of Figure 4.

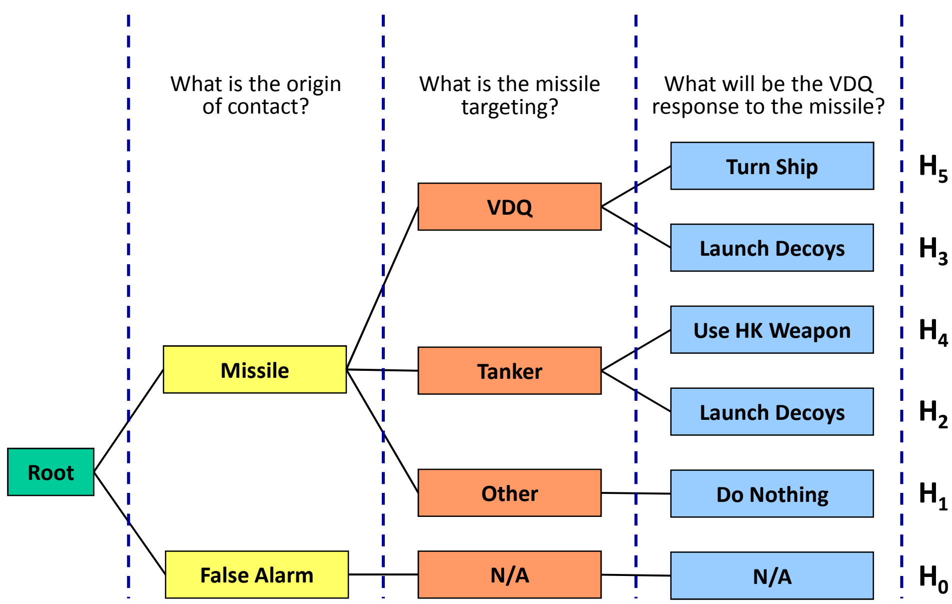



*Figure 7 Hypothesis tree representation*

#### 3.3.1.1 Equivalence of Representations

A key aspect is that the hypothesis tree graphical representation (Figure 7) of the possible situation *must be equivalent* to the “bubbles and links” graphical representation (Figure 4). The two representations must tell the same story. Moreover, this is totally disconnected from any uncertainty model (e.g., probabilities in a Bayesian framework) that could be used (later) to express preferences on the different possibilities. Hence, the two representations (Figure 4 and Figure 7) can be entirely constructed without one having to care about probabilities at all.

### 3.3.1.2 Containers Without Semantics

Another very important aspect is that the situation model components of types *element* and *relation* in Figure 5 are only « place holders » or « containers ». As such, they don't by themselves convey any particular semantics related to the situation being modeled. It is the actual contents of the situation model components that make sense (or not) to human analysts. The MHSA support system manipulates the place holders (or containers), not the contents. Hence the system doesn't care about the semantics of the contents. For the system, these contents are totally irrelevant. This is illustrated in Figure 8.

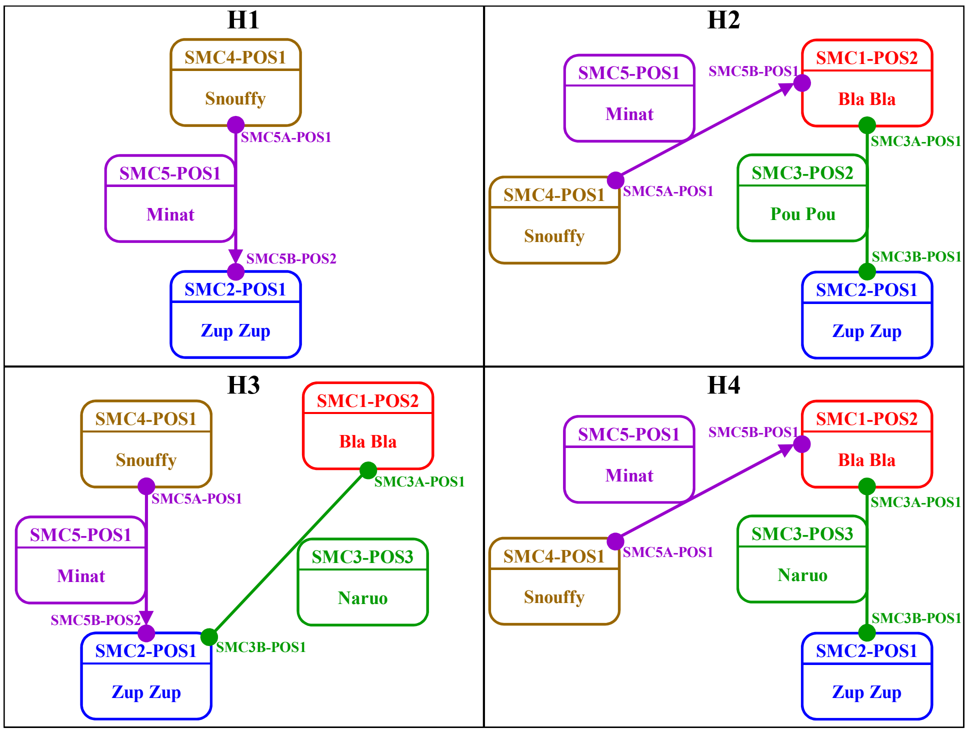



*Figure 8 Containers with meaningless contents*

There is certainly a semantics related to the graphical language itself. For example, the MHSASS understands the meaning of what the "origin of a relation" is from a graph point of view, and what it is allowed (or not) to do with this component from a "container management" perspective, but the support system doesn't understand the meaning of the actual contents of any SMC.

This is an important aspect, as the MHSASS can be used in different domains that make sense to the user but that are totally irrelevant for the system itself. One can thus use the support system to describe a «guest and cooking situation», a «maritime drug smuggling situation», an «improvised explosive device situation», etc.

Figure 9 shows the hypothesis tree matching the hypothetical part of the situation modeled in Figure 8. Note that SMC002, SMC003A, SMC003B, SMC004, SMC005 and SMC005A are all *certain* components in Figure 8. As such, they are present in all hypotheses of Figure 8 and Figure 9.

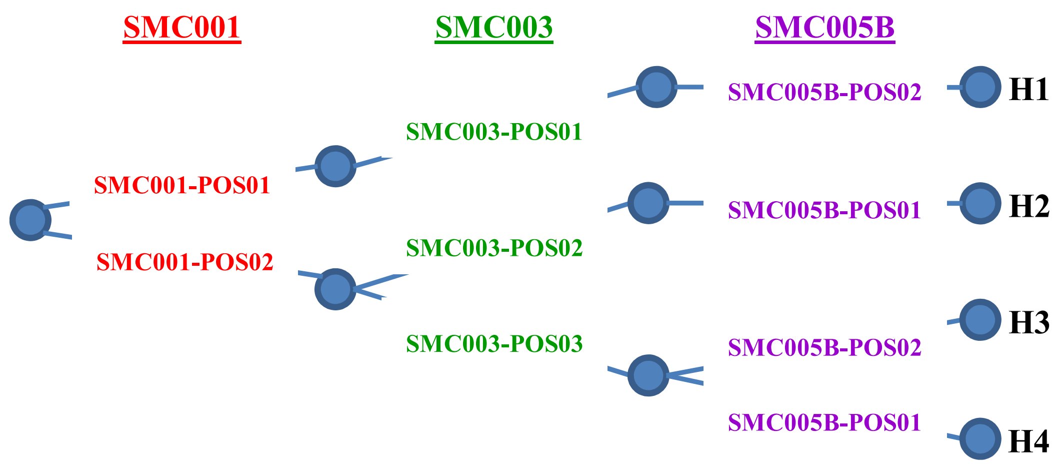



*Figure 9 Hypothesis tree with meaningless containers*

#### 3.3.1.3 Component indexing mechanism

There has to be an indexing mechanism to identify the SMCs and their possibilities, which must be totally independent of any semantics related to the situation. A tag such as *SMC004-POS07* is used to identify *possibility #7* for the *situation model component #4*. For convenience, the “not existent” possibility for a component is always tagged as *POS00*. Three tags are necessary to identify a relation: *SMC003-POS02*, *SMC003A-POS01*, *SMC003B-POS05* are used to identify *possibility #2* for the relation *SMC #3*, with *possibility #1* for its origin, and *possibility #5* for its destination.

### 3.3.2 Tree Construction

In the beginning, the MHTree is represented by a single (root) node. The MHTree is built by iteratively adding (one or multiple) nodes to the tree for every SMC (one tree level per SMC). The number of nodes that can be added to the tree for a given SMC strictly depends on the uncertainty, or the possibilities, associated to this component. The manner in which they are added depends on the possibility dependency requirements of the possibilities for the SMC. Figure 10 shows an example of a generic hypothesis tree.

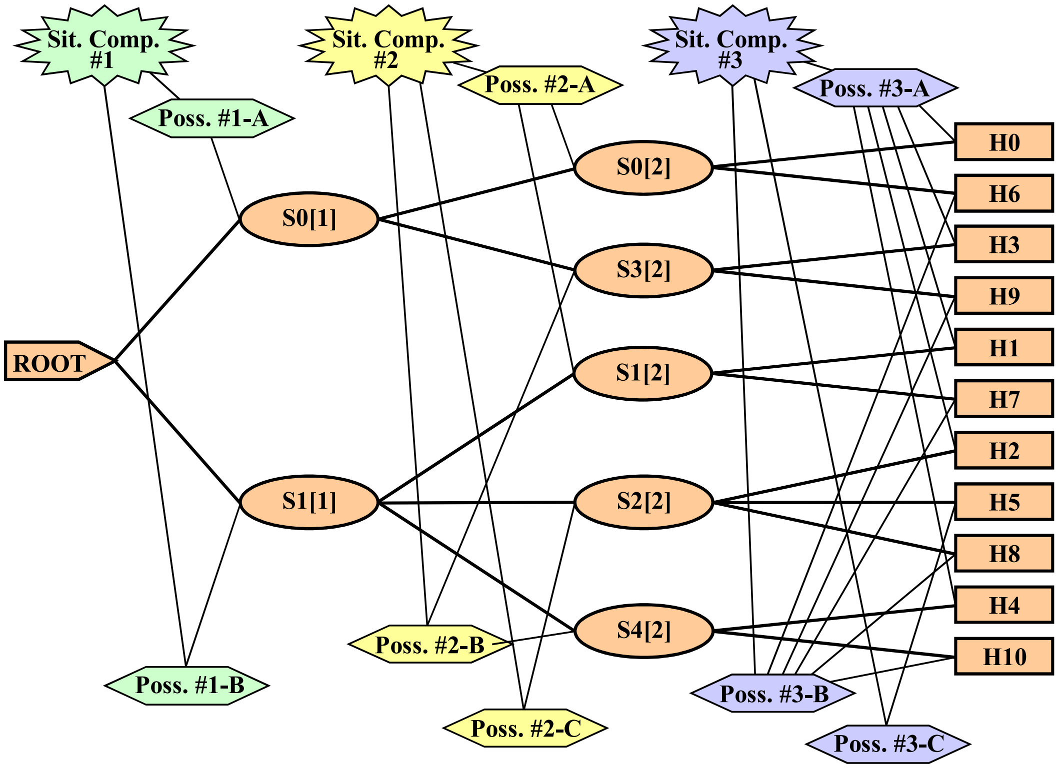



*Figure 10 Multiple Hypothesis Tree Construction*

The root node is connected to various SMC possibilities. Every SMC (Sit. Comp#1, Sit. Comp#2, Sit. Comp#3) has it's possibilities (Poss #x-y) shown on a distinct level of the tree. Finally, each hypothesis is displayed on the right (H0 to H10).

#### 3.3.2.1 Certain Situation Model Components

Certain situation model components, those which are considered to definitely exist in the real world and are common to all hypotheses, are added "before" the root node. They are represented by a single node, and are linked in a "chain-like" manner to the root node. Figure 11 shows such nodes as "starred" nodes.

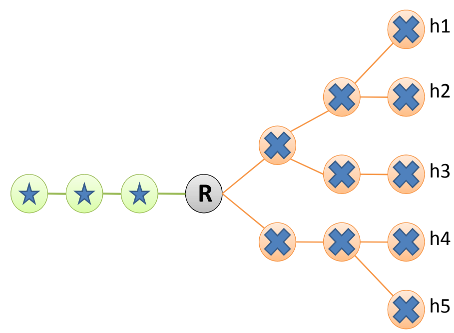



*Figure 11 MHTree example*

#### 3.3.2.2 Uncertain Situation Model Components

Each uncertain situation model component is represented by a "level" of the tree. Figure 7 shows an example of this, as each uncertain SMC (origin of the contact, what is the missile targeting, what will be the VDQ response to the missile) is represented by a different level of the tree. Each SMC possibility is represented by at least a node. Figure 11 shows such nodes as "crossed" nodes.

For every uncertain SMC, at least one new node is added after every leaf node (except for the one at the end of the "certain" components chain). For each SMC possibility, a node is added after every leaf node that represent hypothetical situations which contain all the elements that the new SMC possibility impacts. The construction of hypothetical situations (hypotheses) is discussed in subsection 3.3.2.3. The term "impacts" refers to possibilities which have a structural dependency with one another, i.e., that either "require" or "affect" one another. This is the subject of subsection 3.3.2.5. If a particular leaf node represents a hypothetical situation which does not contains the elements that the new SMC possibility impacts, a dummy node (N/A in the example of Figure 7) is added. If a new SMC possibility does not impact any of the other SMC, it will be added after every leaf node.

##### 3.3.2.2.1 Clustering

The addition of uncertain components will be different if clustering of SMC is used. Clustering allows the creation of distinct hypothesis trees for SMCs that have no dependency between (that do not impact or affect) one another. When possible, having distinct trees reduces the complexity and size of the individual tree data structures.

If clustering is "enabled", the tree expansion process previously described is slightly different. If a new SMC possibility is independent (does not require or affect any other SMC possibility) it will be added to a new, distinct tree.

Note that in some cases, new SMC possibilities may require or affect SMC possibilities from distinct trees of clustered SMC. In such cases, the different trees will be merged, as their respective possibilities will now be dependant.

An in depth description of MHT clustering is availaible in [6]. Cluster formation, merging, splitting and deletion, and complexity is discussed in greater detail.

#### 3.3.2.3 Reconstructing an Hypothesis

A hypothetical situation, or hypothesis, can be reconstructed by starting from a leaf node and tracing back all the possibility nodes up to the root. This process is repeated for every tree, if many trees (of clustered components) are present. Figure 11 shows five such hypotheses, noted h1 to h5.

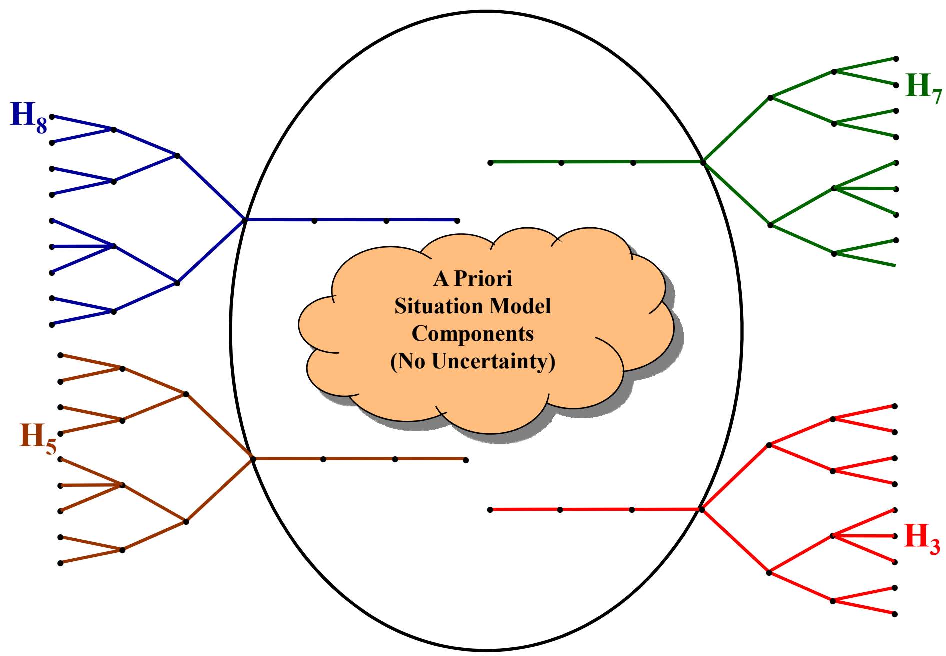



*Figure 12 Reconstructing an hypothesis from clustered trees*

Figure 12 shows an example of 4 clustered trees. A particular hypothesis, for a given hypothetical situation, can be reconstructed by combining distinct hypotheses from every tree. In this particular example: Reconstructed hypothesis $H_{8573}$ is built by combining a branch from every clustered tree $H_8$+$H_5$+$H_7$+$H_3$, each branch representing an hypothesis of a distinct clustered tree.

#### 3.3.2.4 Computing Likelihood

An essential aspect of the MHSA framework is a capability to quantify the degree to which a given hypothesis is “the correct one”. Equipped with such a capability, one can attach a value, i.e., a score, to each individual hypothesis, which is essential for the management of the hypotheses and to ultimately decide on the best output results to be provided to the analyst.

In turn, hypothesis scoring requires uncertainty modeling, and many options can be considered (Bayesian framework, evidence theory, etc.). Whatever the approach selected however, a key issue is that one doesn’t have to care at all about any particular uncertainty model during the construction of the “hypothesis tree” and “bubbles and links” graphical representations; they can be entirely constructed without talking probabilities at all.

For the current version of the MHSASS, Bayesian probabilities are used to compute likelihood. Cumulative hypothesis probability score can be computed by multiplying the conditional probability value for each node of the hypothesis branch. Figure 13 shows an example of this.

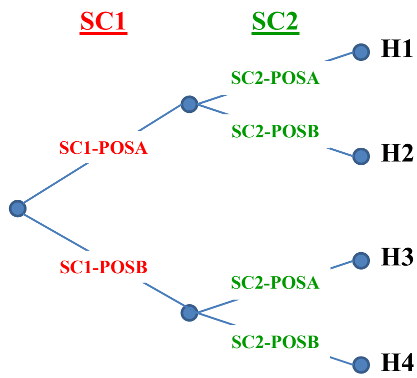



*Figure 13 Structurally independent situation model components*

In this example, we have the following conditional probabilities:

- P(SC2-POSA) = 0.3, while P(SC2-POSB) = 0.7.

“If SC2 is SC2-POSA, there is an 80% chance of having SC1-POSA, and a 20% chance of having SC1-POSB.”.

“If SC2 is SC2-POSB, there is an 10% chance of having SC1-POSA, and a 90% chance of having SC1-POSB.”.

We can compute the probability of each hypothesis (H1 to H4) by multiplying the conditional probability value of possibilities:

P(SM1) = P(H1) = P(SC1-POSA ∩ SC2-POSA) = P(SC2-POSA) P(SC1-POSA | SC2-POSA) = 0.3 x 0.8 = 0.24

P(SM2) = P(H2) = P(SC1-POSA ∩ SC2-POSB) = P(SC2-POSB) P(SC1-POSA | SC2-POSB) = 0.7 x 0.1 = 0.07

P(SM3) = P(H3) = P(SC1-POSB ∩ SC2-POSA) = P(SC2-POSA) P(SC1-POSB | SC2-POSA) = 0.3 x 0.2 = 0.06

P(SM4) = P(H4) = P(SC1-POSB ∩ SC2-POSB) = P(SC2-POSB) P(SC1-POSB | SC2-POSB) = 0.7 x 0.9 = 0.63

where SM1, SM2, SM3 and SM4 are the four possible situation models.

#### 3.3.2.5 Structural Dependency

In the example of Figure 13, the situation model components SC1 and SC2 are structurally independent. That is, if a decision is made for SC2 to keep only the possibility SC2-POSB (for example), then the possibilities for SC1 are not affected; SC1 can still be SC1-POSA or SC1-POSB. Similarly, if a decision is made for SC2 to keep only the possibility SC2-POSA (for example), then the possibilities for SC1 are not affected; SC1 can still be SC1-POSA or SC1-POSB. One can also consider a decision on SC1. If a decision is made for SC1 to keep only the possibility SC1-POSA (for example), then the possibilities for SC2 are not affected; SC2 can still be SC2-POSA or SC2-POSB. Similarly, if a decision is made for SC1 to keep only the possibility SC2-POSB (for example), then the possibilities for SC2 are not affected; SC2 can still be SC2-POSA or SC2-POSB. And this has nothing to do at all with the probabilities.

An example where two situation model components are "structurally dependent" is shown in Figure 14.

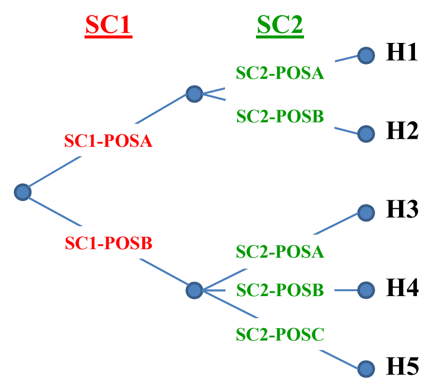



*Figure 14 Structurally dependent situation model components*

In the example of Figure 14, if a decision is made for SC2 to keep only the possibility SC2-POSC, then the only possibility for SC1 is SC1-POSB; it is impossible to have a situation with SC1-POSA if SC2-POSC is selected as the only option for SC2. Similarly, if a decision is made for SC1 to keep only the possibility SC1-POSA, then the possibility for SC2 to be SC2-POSC is eliminated.

The information on structural dependencies is used:

a) during the expansion of the hypothesis tree, when processing a new situation model component, to determine if branching from an existing branch with one particular possibility of the new component is allowed.

b) manage the tree when a situation model component is removed (with all of its possibilities), or more simply when a given possibility is removed.

Situation model components that are "structurally independent" can be clustered, or represented in separate hypothesis trees (3.3.2.2.1).

### 3.3.3 Multiple Hypothesis Tree Construction Example

As mentioned previously, the MHSA system manipulates the place holders (or containers), not the contents. Hence the MHSA system doesn't care about the semantic of the contents. For the MHSA system, the contents are totally irrelevant. However, to illustrate the MHTree construction process, let us look at, and construct the hypothesis tree for, a very mundane situation evaluation problem, using simple semantics. The general scenario goes as follows. Alex is planning tonight's dinner. He is unsure whether he will cook chicken or beef. He is also unsure as to whether he will have a guest. There are two uncertain components in this situation. What will Alex cook? and, will Alex have a guest?

#### 3.3.3.1 Independent Case

For the independent case, let us assume there the following probabilities:

0.8 chance of having a guest, 0.2 chance of having no guest,

0.1 chance of cooking beef, 0.9 chance of cooking chicken.

Figure 15 shows different views of the situation.

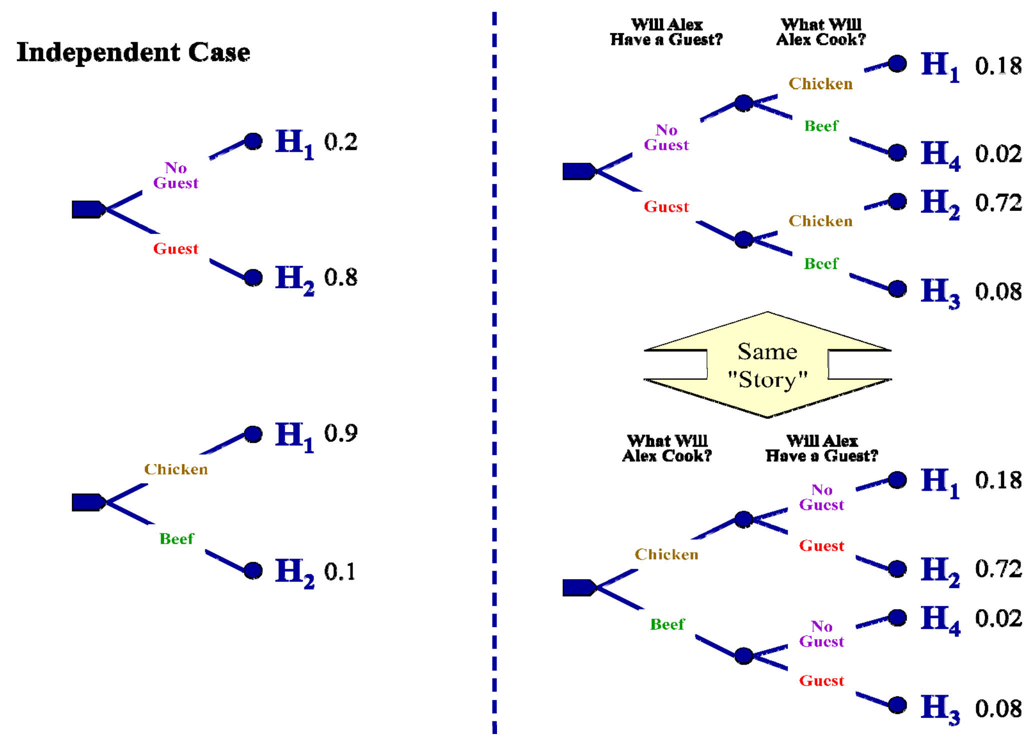



*Figure 15 MHtree construction example, independent case*

Both SMCs are shown on the left with their respective probability. On the right, one can notice that the two hypothesis trees tell the same "story", or reflect the same situation. Each possible hypothesis is present on both trees. These hypotheses are shown on Figure 16.

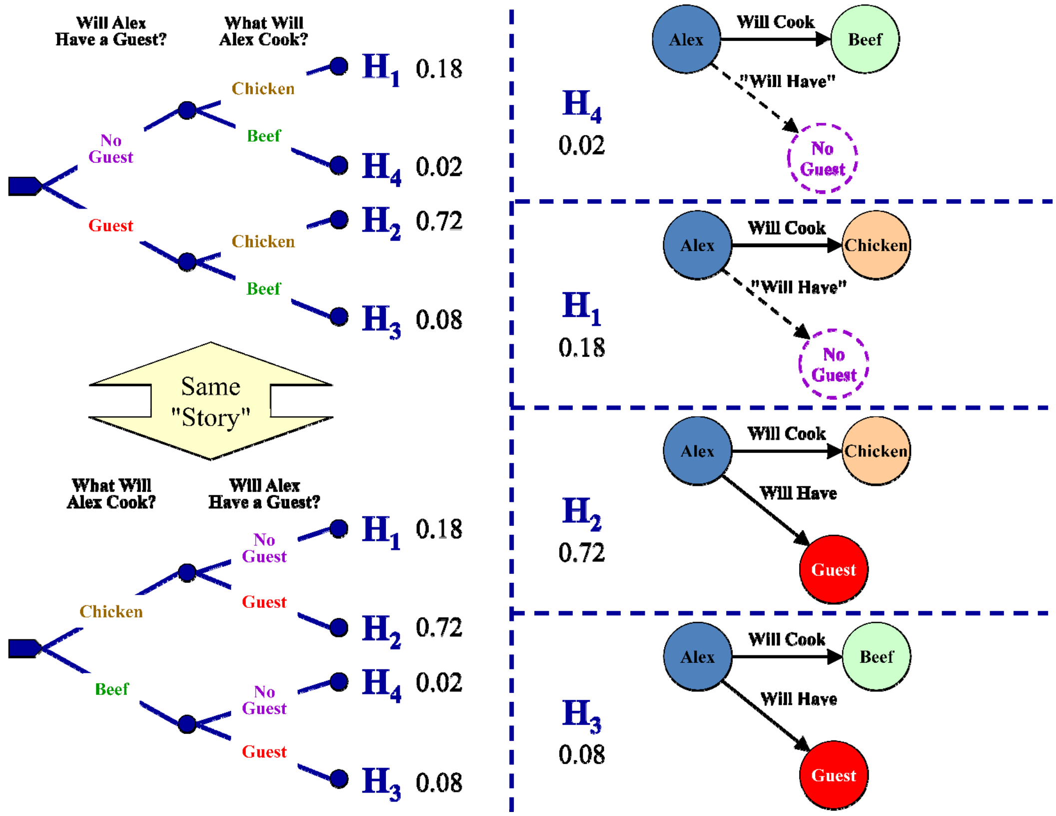



*Figure 16 MHTree construction example, hypotheses*

### 3.3.3.2 Independent Case with Conditional Probabilities

Let us now assume that the menu will change depending on the presence or absence of a guest. The following conditional probabilities now apply:

P (Guest) = 0.2

P (No Guest) = 0.8

P (Chicken|Guest) = 0.01

P (Beef |Guest) = 0.99

P(Chicken|No Guest) = 0.9

P(Beef|No Guest)=0.1

Figure 17 shows different views of the situation.

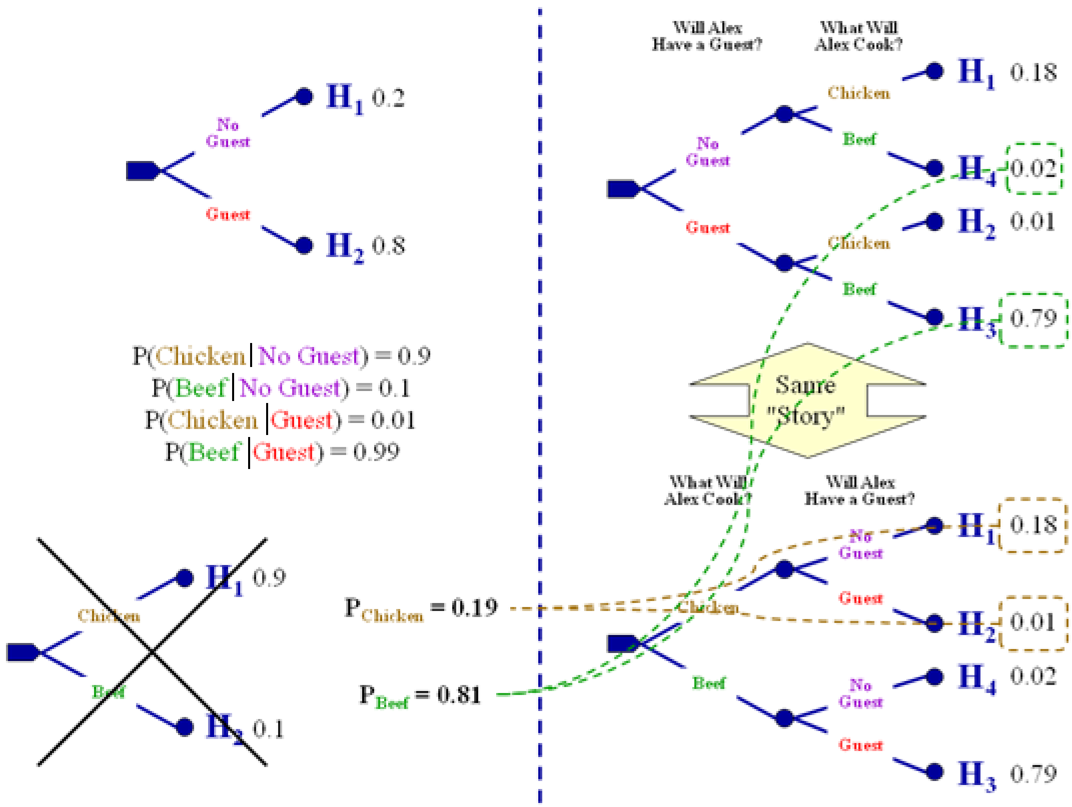



*Figure 17 MHTree construction example, conditional probabilities*

Both SMCs are shown on the left with their respective probability. On the right, one can notice that the two hypothesis trees tell the same "story", or reflect the same situation. Each possible hypothesis is present on both trees. These hypotheses are shown on Figure 18.

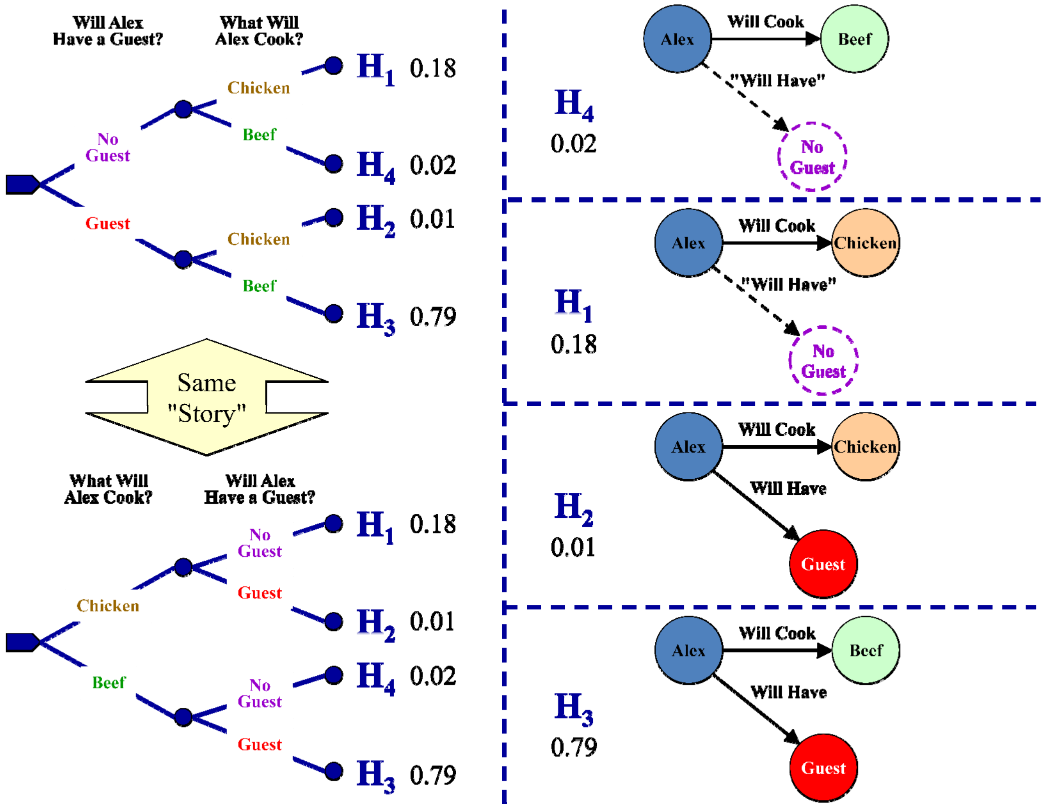



*Figure 18 MHTree construction example, hypotheses- conditional probabilities, hypotheses*

The hypothesis probability score is computed using the conditional probability formula $P(A \cap B) = P(B)P(A|B) = P(A)P(B|A)$. The reverse conditional probabilities (P(Guest|Chicken)) are computed using Bayes' theorem: $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$.

Note that so far, the components could have been presented, using clustering, on distinct trees, as they have no structural dependency. The next part of this example introduces structural dependency.

### 3.3.3.3 Dependent Case

Let us now consider a variation where the absence of a guest affects (automatically implies) cooking chicken. The following conditional probabilities now apply:

P (Guest) = 0.2

P (No Guest) = 0.8

P (Chicken|No Guest) = 1

P (Beef|No Guest) = 0

P (Chicken|Guest) = 0.01

P (Beef|Guest) = 0.99

Figure 19 shows different views of the situation.

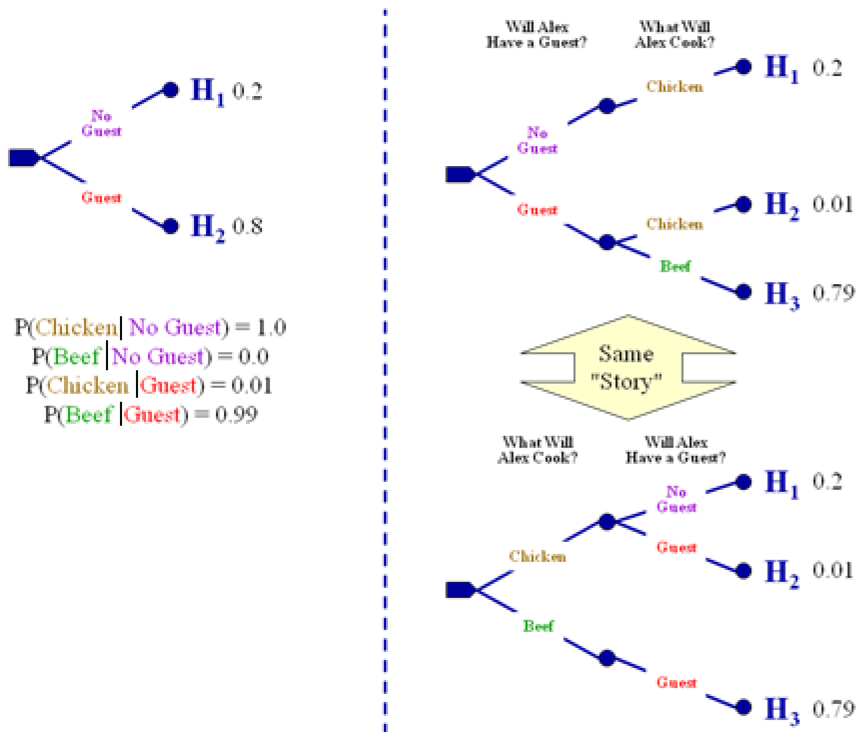



*Figure 19 MHTree construction example, dependent case*

Probabilities are shown on the left. On the right, one can notice that the two hypothesis trees tell the same "story", or reflect the same situation. Each possible hypothesis is present on both trees. In this specific case, however, as cooking chicken is affected by the presence of a guest (P(chicken|non guest)=1), the hypothesis trees reflect the structural dependency. There is no need for a branch showing the no guest/beef option as it has no chance of occurring. The resulting hypotheses are shown on Figure 20. This example provides a good example of the relationship that exists between the "affect" and "require" dependencies. It was stated that the absence of a guest implies cooking chicken. The corollary of this statement is that the absence of a guest also implies not cooking beef, as cooking chicken and cooking beef are mutually exclusive possibilities of a single component. This type of dependency is the subject of section 3.4.

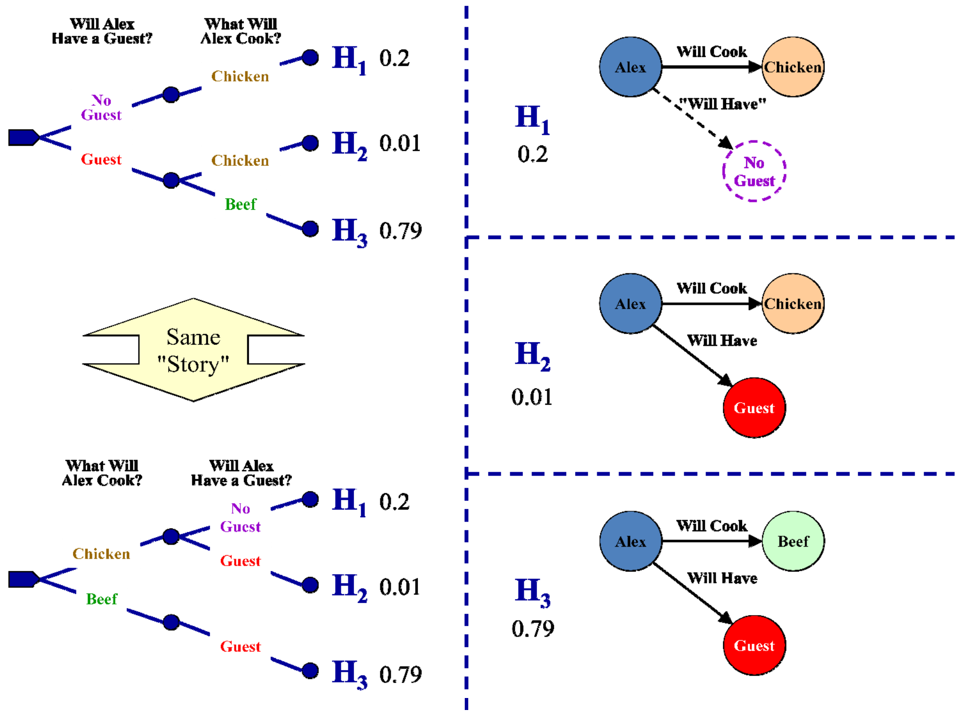



*Figure 20 MHtree construction example, dependent case - hypotheses*

### 3.3.4 A Note on the Representations

Throughout this document, we use different ways to illustrate situations. On Figure 20, for instance, we use both a tree structure and a bubble diagram. Section 4 will introduce the MHSASS prototype and discuss the grid view (or SMC view) of a situation (cf. 4.3.4). It is important to understand that for a given situation, every view reflects and describes the same situation. Every representation tells the "same story", i.e., every possibility of every SMC is present on every view.

## 3.4 Requires/Affects Manager

This section discusses the various intricacies of the SDL in presence of uncertainty and dependency requirements between possibilities. The reader should be aware that it is meant to describe in detail the logical implications of the various SDL constructs. The contents of this section are of a very technical nature, and are intended for individuals seeking a deeper understanding of MHSA: it may not be of interest for more casual readers.

The possibility dependency requirements of the SDL provide the analyst with greater expressivity. It is important to understand that dependency requirements affect SMC possibilities is various ways. In the MHSASS prototype, this is managed automatically by the "Requires/Affects Manager" (RAM) which is specially designed to handle the impact on the hypothesis tree management of "requires" and "affects" dependencies. Let us take a look at the various potential interactions that can occur.

### 3.4.1 Requires True

A possibility of a given SMC may be known to require another possibility of another SMC to be "true" in order to, itself, be considered as "true". The first possibility is then said to "require true" the second one. Let us note possibility 'a' "requires true" possibility 'b' like this:

$$a \Rightarrow b$$

This means that if $a$ "requires true" $b$, then $a$ can only be true if $b$ is also true. $a$ may also be false when $b$ is true, $a$ is necessarily false when $b$ is also false, $a$ cannot be true if $b$ is false. The resulting truth table is shown in Table 1:

*Table 1 "Requires True" truth table*

| $a$ | $b$ | $a \Rightarrow b$ | a Requires True b |
|---|---|---|---|
| T | T | T | $a$ can be true when $b$ is true<br>$b$ must be true when $a$ is true |
| F | T | T | $a$ can be false when $b$ is true<br>$b$ can be true when $a$ is false |
| F | F | T | $a$ must be false when $b$ is false<br>$b$ can be false when $a$ is false |
| T | F | F | |

### 3.4.2 True Affects

A possibility of a given SMC is said to "true affect" another possibility of another SMC if that other possibility "requires true" the first one. Let us note possibility $b$ "true affects" possibility $a$ like this:

$$b \Leftarrow a$$

That is, if $a \Rightarrow b$, then $b \Leftarrow a$. This means that if $b$ "true affects" $a$, then $a$ can only be true if $b$ is true, $a$ may also be false when $b$ is true, $a$ is necessarily false when $b$ is false, $a$ cannot be true if $b$ is false. The resulting truth table is shown in Table 2:

*Table 2 "True Affects" truth table*

| $a$ | $b$ | $b \Leftarrow a$ |
|---|---|---|
| T | T | T |
| F | T | T |

| | | |
|---|---|---|
| F | F | T |
| T | F | F |

### 3.4.3 Implicit Requires True

If a possibility of a given SMC "requires true" a possibility of another SMC which, in turn, "requires true" another one from yet another SMC, the first one is said to "implicitly require true" the last one. We note possibility 'a' "implicitly requires true" possibility 'c' like this:

$$a \underset{implicit}{\Longrightarrow} c$$

That is, if $a \Rightarrow b$, and $b \Rightarrow c$, then $a \underset{implicit}{\Longrightarrow} c$.

Note that the chain of "requires true" is not limited to two. It could have been of any length, leading to many sets of "implicit requires true" along the way to the last possibility. Note also that an "explicit require true" may be considered also "implicit", but not the opposite. This means that if *a* "implicitly requires true" *c*, then *a* can only be true if *c* is true, *a* may be false when *c* is true, *a* is necessarily false when *c* is also false, and *a* cannot be true when *c* is false. The resulting truth table is shown in Table 3:

*Table 3 "Implicit Requires True" truth table*

| $a$ | $c$ | $a \underset{implicit}{\Longrightarrow} c$ |
|---|---|---|
| T | T | T |
| F | T | T |
| F | F | T |
| T | F | F |

### 3.4.4 Implicit True Affects

If a possibility of a given SMC is said to "implicitly true affect" another possibility from another SMC, it is "required true implicitly" by this other possibility. Let us note possibility *c* "implicitly true affects" *a* like this:

$$c \underset{implicit}{\Longleftarrow} a$$

This means that if *c* "implicitly true affects" *a*, then *a* can only be true if *c* is true, *a* may be false when *c* is true, *a* is necessarily false when *c* is also false, and *a* is cannot be true when *c* is false. The resulting truth table is shown in Table 4:

*Table 4 "Implicit True Affects" truth table*

| $a$ | $c$ | $c \underset{implicit}{\Longleftarrow} a$ |
|---|---|---|
| T | T | T |
| F | T | T |
| F | F | T |
| T | F | F |

### 3.4.5 Requires False

It may happen that a possibility requires another possibility to be false for itself to be considered as true. We say that the first possibility has a "require false" relationship with the second possibility. Let us note possibility *a* "requires false" possibility *b* like this:

$$a \underset{X}{\Rightarrow} b$$

This means that *a* can only be true if *b* is false, *a* may also be false when *b* is false, *a* is necessarily false when *b* is true, and *a* cannot be true if *b* is true. The resulting truth table is shown in Table 5:

*Table 5 "Requires False" truth table*

| $a$ | $b$ | $a \underset{X}{\Rightarrow} b$ | a Requires False b |
|---|---|---|---|
| T | T | F | |
| F | T | T | *a* must be false when *b* is true<br>*b* can be true when *a* is false |
| F | F | T | *a* can be false when *b* is false<br>*b* can be false when *a* is false |
| T | F | T | *a* can be true when *b* is false<br>*b* must be false when *a* is true |

### 3.4.6 False Affects

When a possibility "requires false" another possibility, that other possibility is said to "false affect" the first one. Let us note possibility *b* "false affects" possibility *a* like this:

$$b \underset{\text{X}}{\Leftarrow} a$$

This means that *a* can only be true if *b* is false, *a* may also be false when *b* is false, *a* is necessarily false when *b* is true, and *a* cannot be true if *b* is true. The resulting truth table is shown in Table 6:

*Table 6 "False Affects" truth table*

| $a$ | $b$ | $b \underset{\text{X}}{\Leftarrow} a$ |
|---|---|---|
| T | T | F |
| F | T | T |
| F | F | T |
| T | F | T |

### 3.4.7 Implicit Requires False

#### 3.4.7.1 Implicit Requires False on Other "Target" Possibilities

If a possibility *a* "requires true" (or implicitly requires true) a possibility $b_0$ for a given SMC *C*, then possibility *a* is said to "implicitly require false" all other possibilities $b_{i(i \neq 0)}$ of this same SMC. This is a consequence of the fact that the possibilities for a given SMC must be mutually exclusive. Figure 21 illustrates this concept.

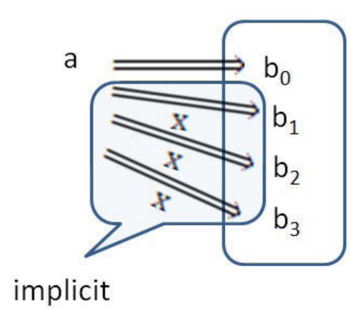



*Figure 21 Implicit requires false on "target" possibilities*

#### 3.4.7.2 Implicit Requires False on Other "Source" Possibilities

If a possibility *a* of SMC#1 "requires true" (or implicitly requires true) a possibility $b_0$ for another component SMC#2, then possibility *a* is said to "implicitly require false" any other possibility of any other SMC that requires true (implicitly or not) any of the other possibilities $b_{i(i \neq 0)}$ of SMC#2. Figure 22 illustrates this concept.

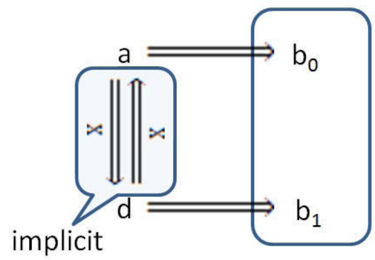



*Figure 22 Implicit requires false on "source" possibility*

#### 3.4.7.3 Implicit Requires False inherited from "required" possibility

If possibility *a* "requires true" possibility *b*, which in turns "requires false" possibility *c*, then possibility *a* is said to "implicitly require false" possibility *c*. Figure 23 illustrates this concept.

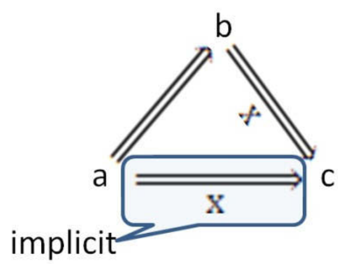



*Figure 23 Implicit requires false inherited from required possibility*

#### 3.4.7.4 Requires False Symmetry

When a possibility *a* "requires false" another possibility *b*, that other possibility can also be said to "implicitly requires false" the first one. Table 7 demonstrates this fact.

*Table 7 Requires false symmetry truth table*

| $a$ | $b$ | $a \underset{x}{\Rightarrow} b$ | $b \underset{x}{\Rightarrow} a$ |
|---|---|---|---|
| T | T | F | F |
| F | T | T | T |

| | | | |
|---|---|---|---|
| F | F | T | T |
| T | F | T | T |

The same can be said for "implicit requires false". Let us then introduce a new notation for "requires false" and "implicit requires false".

$a \underset{\text{X}}{\Leftrightarrow} b$, for "requires false", and

$a \underset{\text{Xi}}{\Leftrightarrow} b$, for "implicit "requires false".

#### 3.4.7.5 Reasoning Implication of Requires False - Part 1

If a possibility *a* of a given SMC "requires false" (explicitly, implicitly, or both) all possibilities $b_i$ of another SMC except for one possibility $b_0$, then possibility *a* is said to "implicitly requires true" that specific possibility $b_0$. Figure 24 illustrates this concept.

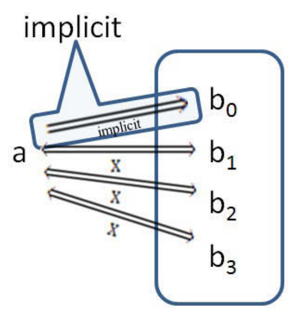



*Figure 24 Reasoning implication of requires false (1)*

#### 3.4.7.6 Reasoning Implication of Requires False - Part 2

If a possibility *a* "requires false" (explicitly, implicitly, or both) a subset of possibilities $\{b_1, b_2\}$ from a given component *B*, then possibility *a* inherits any "requires true" or "requires false" common to the complement set $B\backslash\{b_1,b_2\}=\{b_3,b_4\}$ of component *B* possibilities. Figure 25 illustrates this concept.

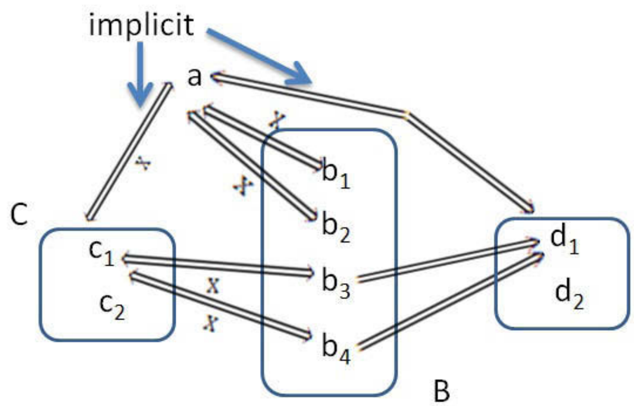



*Figure 25 Reasoning implication of requires false (2)*

#### 3.4.7.7 Reasoning Implication of Requires False - Part 3

If all possibilities $b_i$ of a component *B* except one ($b_3$) require true all possibilities ($c_i$) of another component *C* except one ($c_3$), then the “last” possibility ($c_3$) of the later component *C* implicitly requires true the "non-requiring" possibility $b_3$ of component *B*. The following equations and Figure 26 prove and illustrate that concept.

$b_1 \Rightarrow c_1$ implies $b_1 \underset{X}{\Leftrightarrow} c_2$, and $b_1 \underset{X}{\Leftrightarrow} c_3$ (3.4.7.1)

$b_2 \Rightarrow c_2$ implies $b_2 \underset{X}{\Leftrightarrow} c_1$, and $b_2 \underset{X}{\Leftrightarrow} c_3$ (3.4.7.1)

$b_1 \underset{X}{\Leftrightarrow} c_3$, and $b_2 \underset{X}{\Leftrightarrow} c_3$ imply $c_3 \Rightarrow b_3$ (3.4.7.4 and 3.4.7.5)

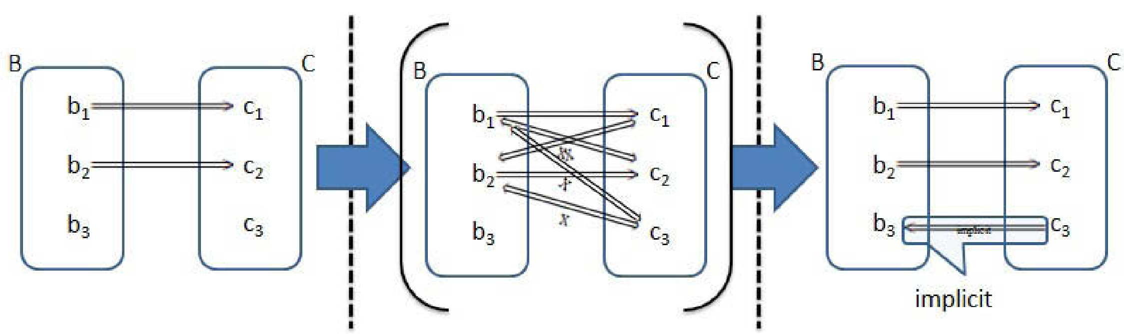



*Figure 26 Reasoning implications of requires false (3)*

### 3.4.8 Relations Implication Inheritance

It is essential to consider various intricate considerations when adding a relation component to the situation model. A relation component is actually composed of three components: the relation itself (core component), the relation origin and the relation destination (cf. 3.2). Each of these components may have multiple possibilities. The core component may have existence and/or content uncertainty, while the origin/destination components may have a uncertainty reflected by the content of the element they represent. Figure 27 shows an example of this.

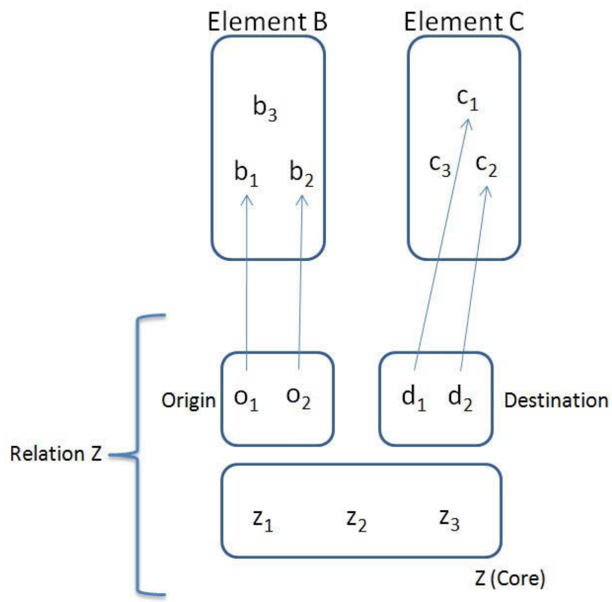



*Figure 27 Relations implication inheritance*

In this example:

relation $Z$ has core possibilities $\{z_1, z_2, z_3\}$, two origin possibilities $\{o_1, o_2\}$, and two destination possibilities $\{d_1, d_2\}$;

the $o_1$ origin possibility links to the $b_1$ value possibility of element $B$;

the $o_2$ origin possibility links to the $b_2$ value possibility of element $B$;

the $d_1$ destination possibility links to the $c_1$ value possibility of element $C$;

the $d_2$ destination possibility links to the $c_2$ value possibility of element $C$.

Defining origin possibility $o_1$ to point at $b_1$ causes $o_1$ to implicitly requires true $b_1$.

Defining origin possibility $o_2$ to point at $b_2$ causes $o_2$ to implicitly requires true $b_2$.

Defining destination possibility $d_1$ to point at $c_1$ causes $d_1$ to implicitly requires true $c_1$.

Defining destination possibility $d_2$ to point at $c_2$ causes $d_2$ to implicitly requires true $c_2$.

Now, let $z_3$ be a "does not exist" possibility. In such a case, the origin and destination components do not exist either. Let $o_3$ and $d_3$ be the "does not exist" possibility for the origin and destination components of $Z$. Figure 28 shows the additions.

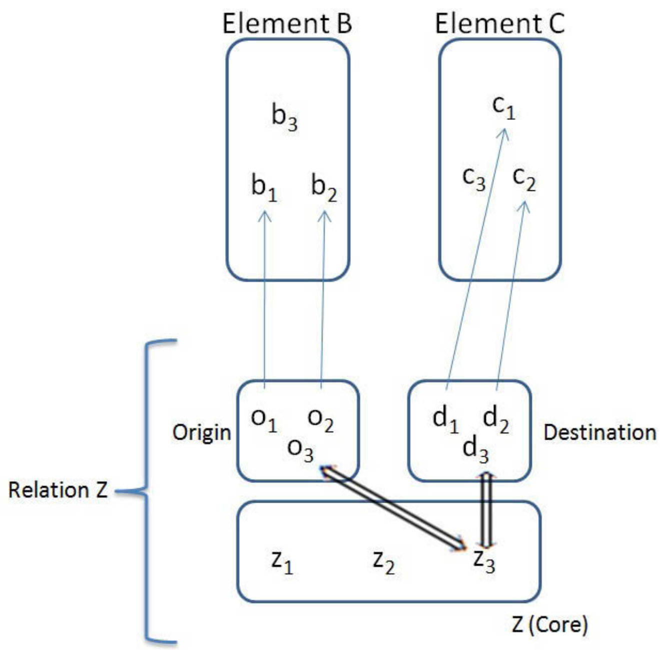



*Figure 28 Relations implication inheritance (2)*

Both $o_3$ and $d_3$ "require true" et "true affect" $z_3$. The presence of relation $Z$ therefore creates the following additional require/affect interdependencies:

$o_1 \Rightarrow b_1$, $o_2 \Rightarrow b_2$, $d_1 \Rightarrow c_1$, $d_2 \Rightarrow c_2$,

$o_1, o_2, d_1, d_2, z_1, z_2, \underset{X}{\Leftrightarrow} o_3, d_3, z_3$

therefore, if $b_3$ or $c_3$ is true, the relation $Z$ cannot exist.

$o_3 \underset{implicit}{\Longrightarrow} z_3$, and $d_3 \underset{implicit}{\Longrightarrow} z_3$.

### 3.4.9 Relation Loop Back

The "loop back" option allows a relation to have the same situation element for its origin and destination. If the “loop back” option is not allowed, then a relation must link two different elements together. On the previous example (Figure 28), all possibilities for the origin are from one element, and all possibilities for the destination are from another element. But there is no such limitation in the SDL. Origin and destination possibilities could be on the possibilities of the same element. Figure 29 shows an example of this concept.

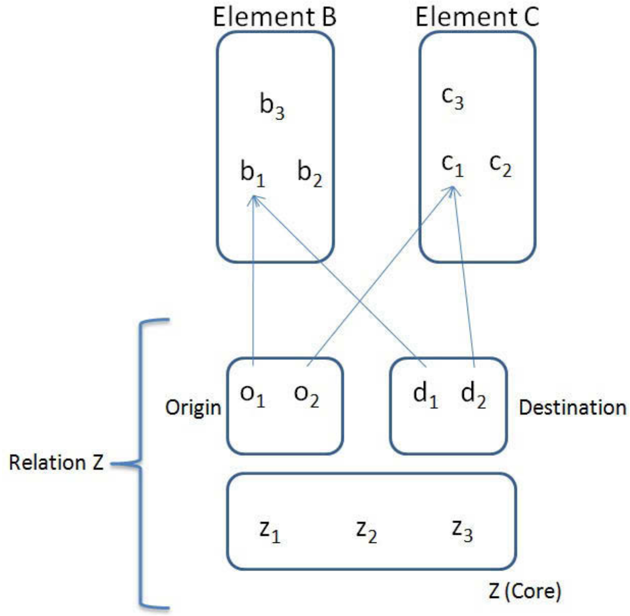



*Figure 29 Loop back implications*

In this example, both the origin and the destination can be possibility $b_1$ or $c_1$. But they cannot be the same possibility at the same time if a relation is not allowed to have the same element as its origin and its destination on a given hypothesis (allowed a "loop back"). There must be some "requires false" linking competing origin and destination possibilities when "loop back" is not allowed. In this example, if "relation loop back" is not allowed:

$o_1 \underset{X}{\Leftrightarrow} d_1$, and $o_2 \underset{X}{\Leftrightarrow} d_2$

### 3.4.10 Sibling Rivalry

In accordance with the previous subsection, if there exists a relation *Z*, such that one of its origin possibilities *o1* points at possibility *b1 of* an element (*B*), and one of its destination possibilities *d1* points another possibility *b2* of the same element (*B*). A "requires false" relationship must be added between the origin and destination possibilities $o_1$ and $d_1$ in order to ensure mutual exclusiveness of possibilities. Figure 30 shows an example of this concept.

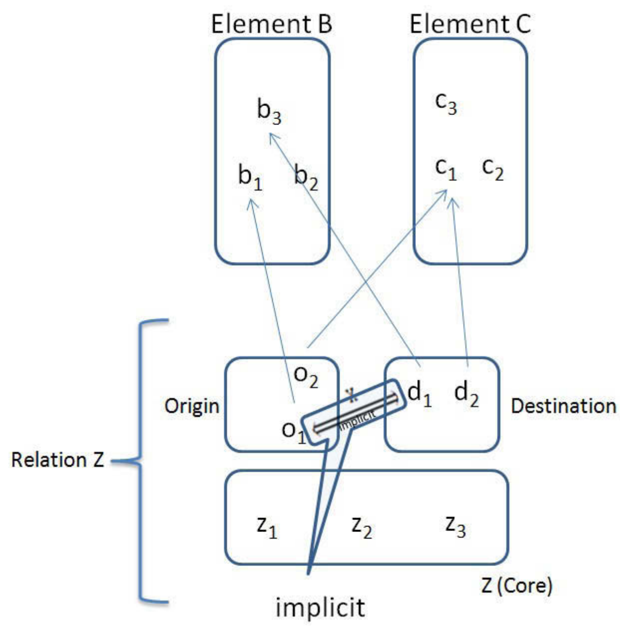



*Figure 30 Sibling rivalry*

### 3.4.11 Dynamic Impact of "Requires" and "Affects"

When new SMC are added to a situation description, the "requires"/"affects" manager (RAM) dynamically handles all the independencies discussed throughout this section. Let us look at a few rules governing the dynamic changes that the RAM implements.

When a component has a single, unique possibility, the component is said to be certain. Its sole possibility is therefore implicitly "required true" by all other possibilities of all other components.

Let $a$ be the sole possibility for component $A$ and $b_0$, $b_1$ the two possibilities for component $B$. If a "requires true" dependency requirement is added between $a_0$ and $b_0$, then $b_1$ must and will be deleted (Figure 31). Possibility $a$ is certain, and as it requires $b_0$, $b_1$ becomes impossible (because of the "mutually exclusive" restriction) and is deleted by the RAM. The corollary is that in order to add a relationship between $a_0$ and $b_0$ without $b_1$ being deleted, one must first add a "does not exist" possibility to $A$.

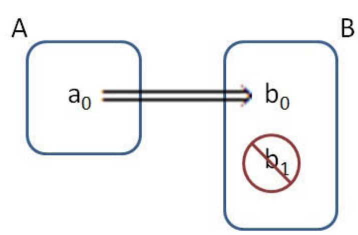



*Figure 31 Requires true causes deletion*

If a relationship $a \Rightarrow b$ exists, the deletion of $b$ invalidates $a$ and causes it to be deleted by the RAM.

If a relationship $a \underset{\mathrm{X}}{\Rightarrow} b$ exists, the deletion of $b$ confirms $a$ (because $a$ is then in all possibilities of the component that used to have $b$ in its possibilities), and potentially invalidates all other possibilities mutually exclusive with $a$.

If a relationship $a \underset{\mathrm{X}}{\Leftrightarrow} b$ exists, where *{b,c}* are possibilities of a component *C,* then the deletion of possibility $c$ invalidates $a$.

Deleting a component means that this component is now considered irrelevant to the situation. It is simply removed by the RAM, ignoring who it may require or affect. Making a component irrelevant implies:

Removing any explicit requires/affects from all the possibilities of this component.

Removing all possibilities of this component

Removing the component

To state that a component “does not exist” one must “edit” the component to ensure the “does not exist” possibility is valid and delete all alternate possibilities. Hence, making the “does not exist” possibility the only possibility of a component is the way to state that this component “does not exist”.

To avoid bringing a relation to an incoherent state, it has been chosen to “prevent” a component deletion for as long as this component is required by a relation origin or destination (i.e., it is targeted as a possible origin and/or a possible destination). If a component is referred to by a relation ending (origin or destination), one must first delete the relation (make it irrelevant) or delete the relation ending (origin or destination) that refers to this component before being allowed to “delete” (make irrelevant) the component.

### 3.4.12 Situation Description Language and Propositional Logic

Propositional logic (PL) is a formal system in which a formal language may be used to represent propositions. A system of inference rules and axioms allows certain formulas to be derived which may be interpreted as true propositions. A thorough look a PL reveals interesting similarities with the SDL. There exists numerous references (for example [11]) detailing propositional logic concepts, operators and axioms. Similarly, Situation Calculus [12] allows the representation of dynamical domains using first order logic. Let us look at a few inference rules that are common between PL and SDL.

#### 3.4.12.1 Modus Ponens

Modus Ponens is possibly the most common inference rule. Let $p$ and $q$ be propositions, modus ponens is a logical implication of the following form: if $p$ implies $q$ and $p$ is true, then $q$ is true. We will note this logical implication as:

$$p \rightarrow q$$

In order for $p \rightarrow q$ to be a valid proposition, $p$ must be false or $q$ must be true. Table 8 shows the resulting truth table.

*Table 8 Logical implication truth table*

| $p$ | $q$ | $p \Rightarrow q$ |
|---|---|---|
| T | T | T |
| F | T | T |
| F | F | T |
| T | F | F |

Referring to Table 1 of Section 3.4.1, it is pretty obvious that the logical implication and the "requires true" dependency requirement share the same truth table values. This similarity is all the more interesting as a number of logical rules can be derived from modus ponens. We will look at a few more PL argument forms, and see how they relate to the SDL.

#### 3.4.12.2 Modus Tollens

Let $p$ and $q$ be propositions, modus tollens is derived from modus ponens, and takes the following form: if $p$ implies $q$ and $q$ is false, then $p$ is false. Once again, this links with Table 1 of Section 3.4.1 and mimics a property of the "requires true" relationship of the SDL.

#### 3.4.12.3 Hypothetical Syllogism

Let $p$ imply $q$ and $q$ imply $r$, hypothetical syllogism dictates that $p$ implies $r$. This transitive property of the logical implication is also present in the SDL. Indeed, 3.4.3 shows that the "implicit requires true" shares the same transitive property.

#### 3.4.12.4 Disjunctive Syllogism

Disjunctive syllogism can be interpreted as: $((p \vee q) \wedge \bar{p} \rightarrow q)$. This reads as: if $p$ or $q$ and not $p$ then $q$. This is somewhat reflected in the SDL by the fact that component possibilities are mutually exclusive and only one possibility must be true for a given component. If we refer to disjunctive syllogism, this would translate by saying, if component $C$ has $a$ or $b$ as possibilities, and possibility $a$ is invalidated, then possibility $b$ is automatically validated. This property can be generalized to more than one component through disjunction associativity, i.e., if component $C$ has $a$ or $b$ or $c$ as possibilities, and possibility $a$ is invalidated, then possibility $b$ or $c$ is automatically validated.

$$\big((p \lor q \lor r) \land \bar{p} \rightarrow (q \lor r)\big)$$

### 3.4.12.5 Constructive Dilemma

Constructive Dilemma can be interpreted as: $(p \rightarrow q) \land (r \rightarrow s) \land (p \lor r) \rightarrow (q \lor s)$. This reads as: if $p$ implies $q$, and $r$ implies $s$, and $p$ or $q$ are true, then $q$ or $s$ are true. This somewhat reflects the principle described in 3.4.7.2. Referring to Figure 22, we have $a \Rightarrow b_o, d \Rightarrow b_1, a \lor d$ which leads to implicitly having. $b_o \lor b_1$.

### 3.4.12.6 Basic Propositional Logic Limitations

Sections 3.4.12.4 and 3.4.12.5 both associate the SDL and PL constructs in imperfect ways. Indeed, the mutual exclusiveness of component possibilities does not perfectly reflect logical disjunction (∨). Indeed, in PL, both values of a disjunction are allowed to be true at the same time, which is not the case for component possibilities in the SDL. The mutual exclusion of possibilities in the SDL is more akin to the mutual exclusive operator or xor (⊕). The following truth table illustrates this fact.

| $p$ | $q$ | $p \lor q$ | $p \oplus q$ | p q |
|---|---|---|---|---|
| T | T | T | F | F |
| F | T | T | T | T |
| T | F | T | T | T |
| F | F | F | F | F |

*Table 9 Disjuction, xor and component possibility truth table*

The xor is not part of propositional logic's basic operators. Using propositional logic, the xor could be expressed as $(a \oplus b) \equiv \left(a \land \bar{b}\right) \lor (\bar{a} \lor b)$ .

Our comparison between the SDL and PL ends here. However, it would be of interest to revisit PL's derived arguments to see how they could be rewritten and validated using xors instead of disjunctions. Succeeding in such an endeavour would allow the construction of a complete mapping of the SDL and PL "allowed" operations. In turn, having such a mapping might allow managing the SDL constructs with an inference engine. This is grounds for potential future works.

# 4 Multiple Hypothesis Situation Analysis Support System Prototype

The Multiple Hypothesis Situation Analysis Support System (MHSASS) prototype is an instantiation of the concepts discussed in Section 3. Figure 32 gives a conceptual view of the system that was implemented. Through "user interactions" (GUI), the user represents his conception of a "true situation". Components of this representation are stored in a database, along with their uncertainty, through "hypothesis and database management". From these elements, an "hypothesis tree" is built. The "uncertainty manager" keeps track and rates particular, uncertain, aspects of the situation. Finally, the various hypothetical situations are displayed back to the user.

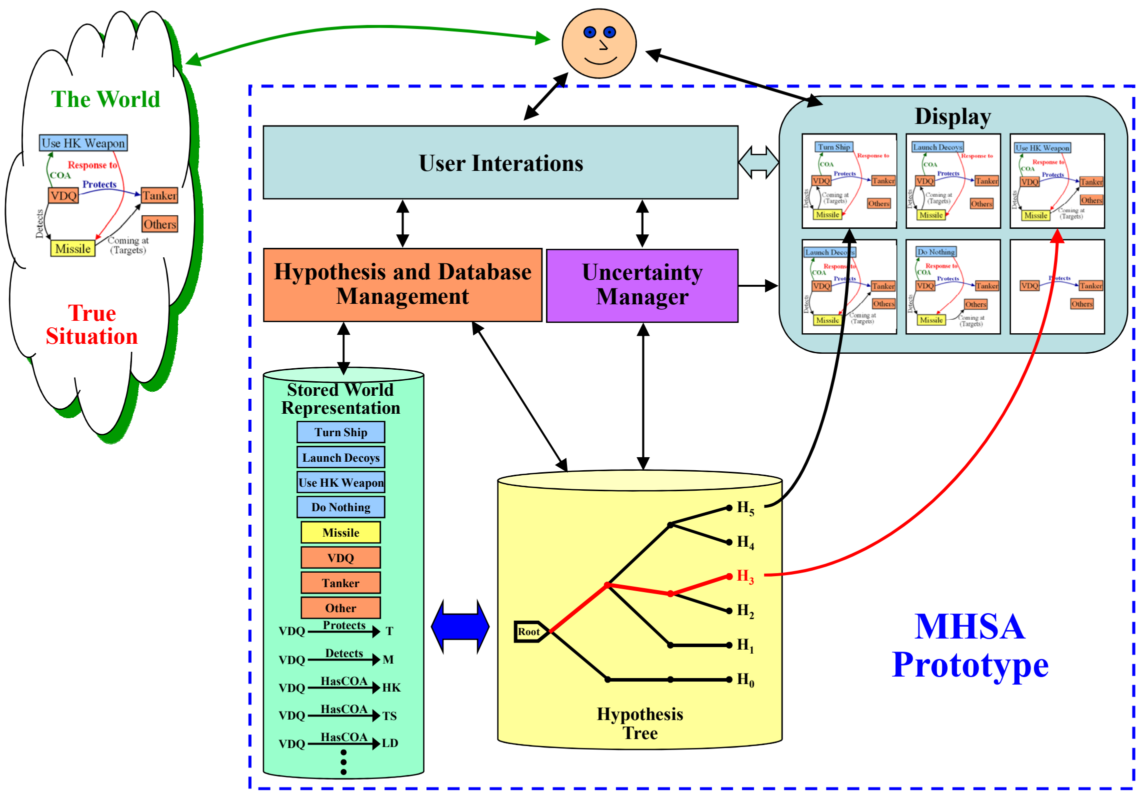



*Figure 32 Conceptual view of the MHSASS*

This section details the steps that were followed in order to develop the MHSA prototype. The following topics are discussed: system requirements, re-use of the Concept Analysis and Simulation Environment for Automatic Target Tracking and Identification (CASE-ATTI) testbed, and prototype overview.

It is worth mentioning that two primary objectives were pursued for this prototype. Obviously, it demonstrates the concept and power of MHSA when put in the hands of the operator, i.e., the added value of helping the user handle multiple hypotheses about an uncertain situation, allowing to defer decisions until new evidence comes in and confirms (or infirms) certain hypotheses. The prototype is also meant as a teaching tool, to help explain the various, sometimes complex, aspects of MHSA.

## 4.1 Prototype Requirements

A series of workshops involving participants from the industry and from DRDC were conducted in order to identify precise requirements for the MHSASS. Following these workshops, a System Requirement Specification (SRS) document was produced detailing the MHSASS requirements.

A list of thirty eight (38) mandatory requirements was described in the SRS. A subset of these requirements is presented here in order to provide a glimpse of the desired MHSASS functionalities:

- Situation Model Interface: *Users shall be able to enter SM components and their various possibilities via a graphical user interface*. This requirement ensures the availability of an easy-to-use interface to allow the description of a situation, its components, and their uncertainty.
- Probabilities Specification: *The user shall be able to enter and change the probabilities for his/her uncertain components*. This requirement allows the user to experiment with the situation models and see how the final situations are affected.
- Hypothesis Tree Display: *The Prototype shall be able to turn on and off the display of the hypothesis tree*. This requirement is meant to give a system developer end user a means to visualize the hypothesis tree, for validation and analysis purposes. It yields information of interest to users with a deep understanding of the MHSA process. It is not useful to the regular, operational type users.
- Hypothesis Visualization: *Visualization means shall be provided to present the situation models and their individual final probability graphically.* The display should show the various hypothetical situations and associated situation characteristics (probabilities), using a graph model diagram.
- Hypothesis Numbering: *Hypothesis numbering and ranking shall be automatically managed.* Every hypothesis generated by the MHSA prototype will be ranked according to its likelihood (for the current prototype, Bayesian probabilities are used to reflect likelihood).

The complete list of system requirements is provided in Annex A.

## 4.2 Potential Reuse of CASE-ATTI for MHSA

One of the tasks during the development of the MHSA prototype was to explore the potential reuse of the existing Multiple Hypothesis Tracking (MHT) implementation from the existing Concept Analysis and Simulation Environment for Automatic Target Tracking and Identification (CASE-ATTI) system. CR 2009-326 [8] describes in detail the results of the investigation.

CASE-ATTI is a multi-sensor data fusion simulation test bed used to analyze the performance of various multi-sensor data fusion architectures and algorithms for the Canadian Patrol Frigate, it was developed by Defence Research & Development Canada – Valcartier (DRDC Valcartier).

The CASE-ATTI system contains data structures and algorithms to perform Multiple Hypothesis Tracking (MHT). The MHT implementation has been revisited in order to provide the MHSA prototype with multiple hypothesis management capabilities [3].

After reviewing all the documentation available and analysing all the hypothesis tree code from CASE-ATTI, the following findings were documented: "The MHT part of CASE-ATTI contains some code that is generic enough to satisfy many requirements from MHSA. However, some of the basic hypothesis tree data structures of MHT are already tainted by specific features such as false alarms and some tracker functions or variables which need to be sorted out. The main classes are too big, for MHSA needs, and should be downsized. The level of effort to mix C++ with Java code would be equivalent or greater to rewrite a pure Java generic multi-hypothesis library. As it is, the hypothesis tree data structure of CASE-ATTI cannot support all the requirements of MHSA but could be used as a starting point to write a new one more suitable for MHSA. The clustering algorithm, although very efficient, is complex and too closely related to tracking. A simpler one could be designed based on dependencies between the nodes. The C++ code used in CASE-ATTI is old and is somewhat outdated with respect to the recent more ANSI compliant C++ compilers." [3].

Based on these findings it was decided not use the source code from the CASE-ATTI. We opted to rewrite a new hypothesis tree Java library closely inspired from a skeleton version of the generic MHT parts of CASE-ATTI.

## 4.3 Prototype Overview

This section provides an overview of the MHSASS prototype. Although not a complete alternative to a live demonstration, it provides a good overview of the MHSA prototype and its functionalities. Understanding of concepts described in Section 3 will be useful to understand some of the functionalities discussed.

### 4.3.1 Main Interface

The MHSA main interface is shown in Figure 33.

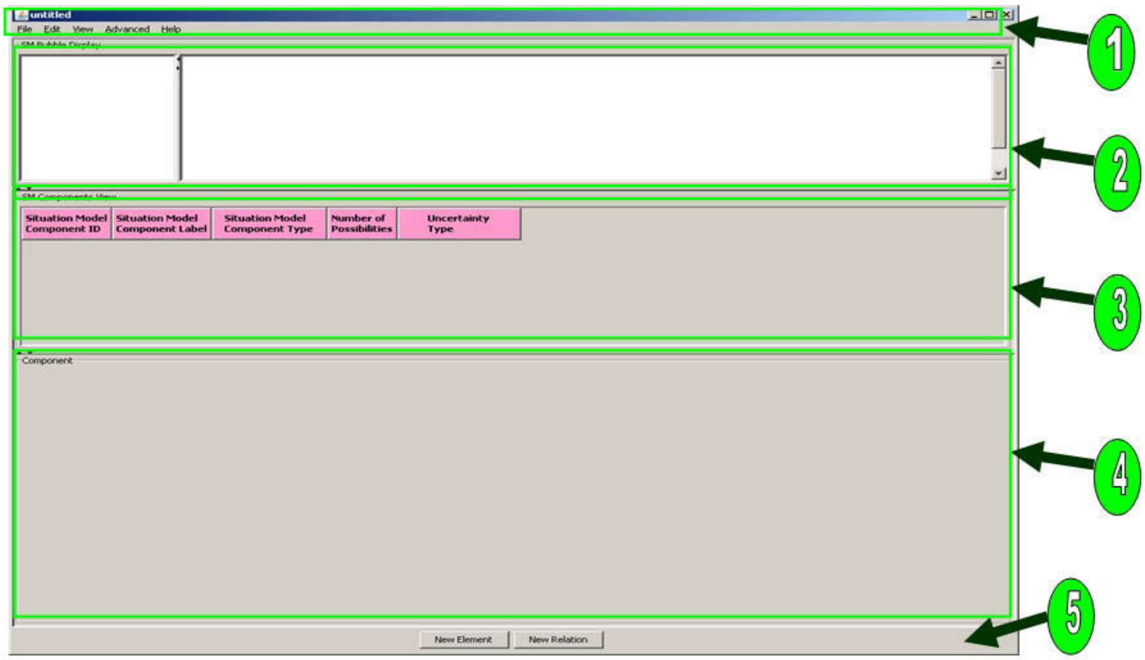



*Figure 33 MHSA main interface*

Figure 33 shows five main items:

Main Menu (File, Edit, View, Advanced, Help)

File (New project, Open project…, Save, Save project…, Quit)

Edit (New Element, New Relation, System options, Display options, Hypothesis management options, Clustering configuration panel)

View (Save Picture as…)

Advanced (View MHTree)

Help (About…)

Situation Model (SM) Bubble Display

SM Component View

Component (detailed viewer/editor)

New Element/Relation (quick) creation buttons

The main menu offers typical application functionalities. The File menu allows users to: create a new project, open an existing project, save a project or quit. The Edit section allows: the creation of a new element, and the creation of a new relation. It gives access to: the system options, the display options, the hypothesis management options, and the clustering and configuration panel. The view menu gives the possibility to save a picture of the current situation. The advanced tab allows the visualization of the hypothesis tree. The help menu provides basic help functions to the user. The SM bubble display is composed of two windows. The right window shows the different hypotheses generated by the system in the form of bubbles (representing elements hence the name) and links representing relations. The left window displays the hypotheses names and their likelihood scores. For the current version of the prototype, likelihood is represented by Bayesian probabilities. The SM component view shows the situation described by the user in the form of a grid. All components (elements and relations) are shown in detail. Every line of the grid contains a particular component with: its ID, its label, its type, its number of possibilities and its uncertainty type. The component editor allows the user to specify in detail every component of the situation being described. Finally, the New Element/Relation buttons give quick access to some of the functionalities available through the edit menu. The more complex functionalities will be discussed in further detail later in this section.

The user may start a new MHSA Prototype project by adding elements or relations or by opening an existing project. At any time, the user may choose to save the current project, either under the current project's name or under a new name. At any time the user may also choose to open a new project and drop all entries made to the current project.

### 4.3.2 Managing an Element Component

#### 4.3.2.1 Adding an Element

To add a new situation element to the situation model, the user clicks on the "Add Element" on the MHSA main frame (refer to Figure 33). This brings up the Element Properties panel shown in Figure 34.

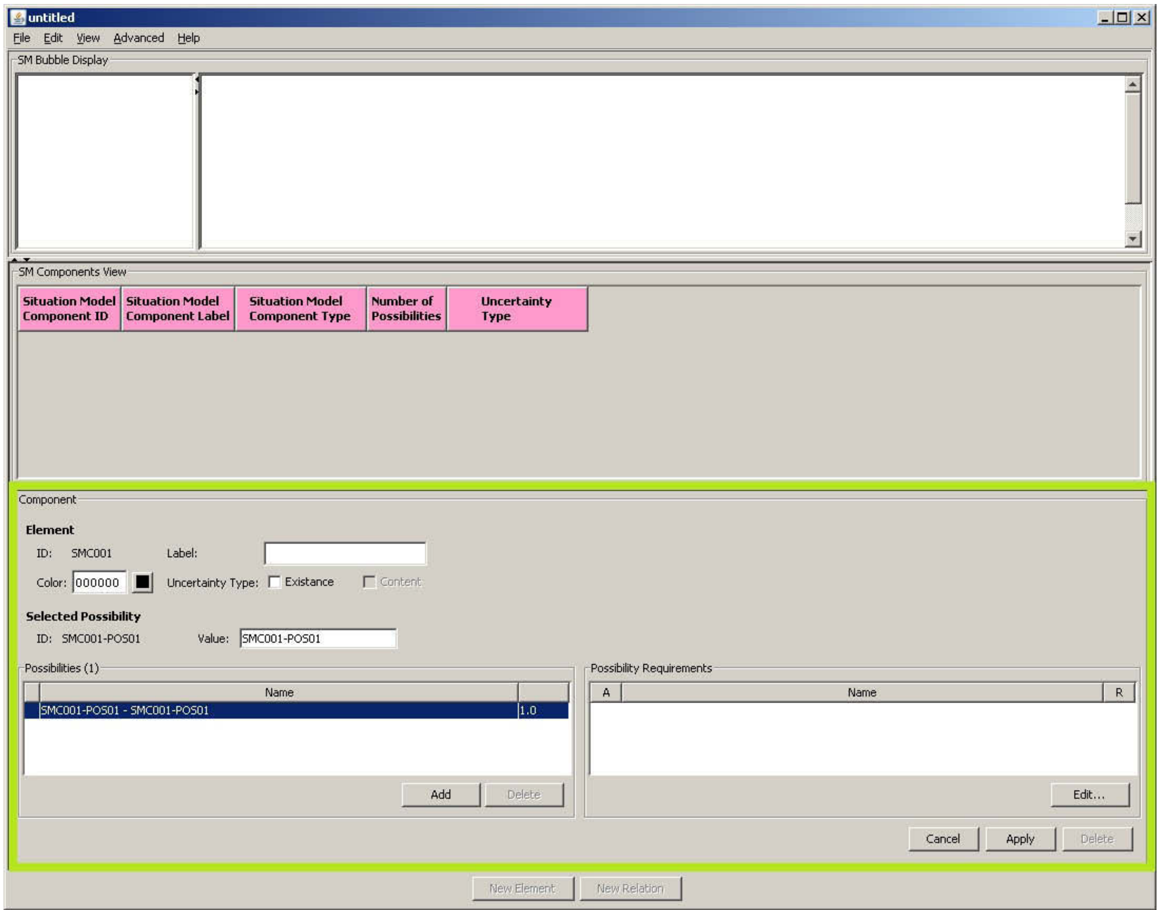



*Figure 34 Element properties panel*

The element identification (ID) is automatically generated to ensure that each SM component has a unique identification. The color item will be used when drawing the element on the SM Bubble Display. The user can see that there is already a possibility shown in the "Possibilities" list since an element must have at least one possibility. The IDs and default values of the possibilities an element are automatically generated. The user is able to change the possibilities default values just by clicking on the value field and entering the desired value string. The user is able to add new possibilities by clicking on the "Add" button shown under the possibility list.

The checkbox "Existence" of the field "Uncertainty Type" can be selected to add a possibility of non-existence for the element. It can be removed by unselecting the existence checkbox. It can also be removed the same way as any other possibilities, by a click on the "Delete" button below the list. If the operation affects other possibilities, a side effect notification will possibly be displayed, depending on the value of the notification settings, as shown by Figure 35.

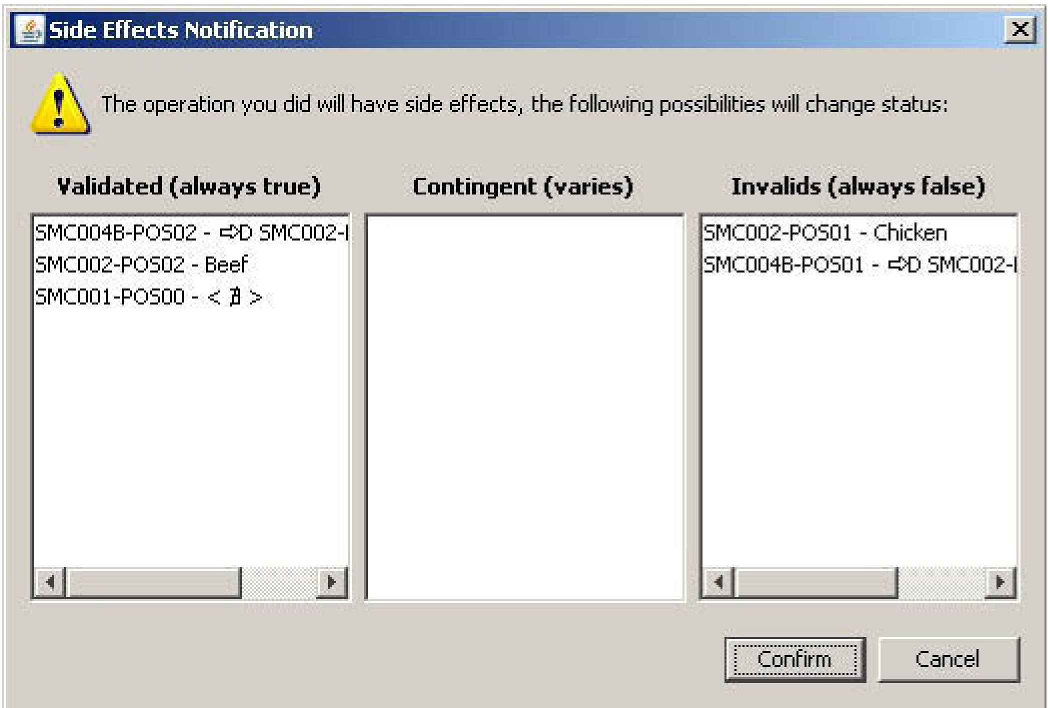



*Figure 35 Side effect notification*

A notification is displayed when a possibility becomes invalid as a result of a modification. It can also be displayed for possibilities that become «validated (always true), or contingents (varies).

If the user wants the current element to "require" some other component possibilities, he/she clicks on the "Edit…" button located under the "Possibility Requirements" list. This brings the "Requires Editor" panel shown in Figure 36.

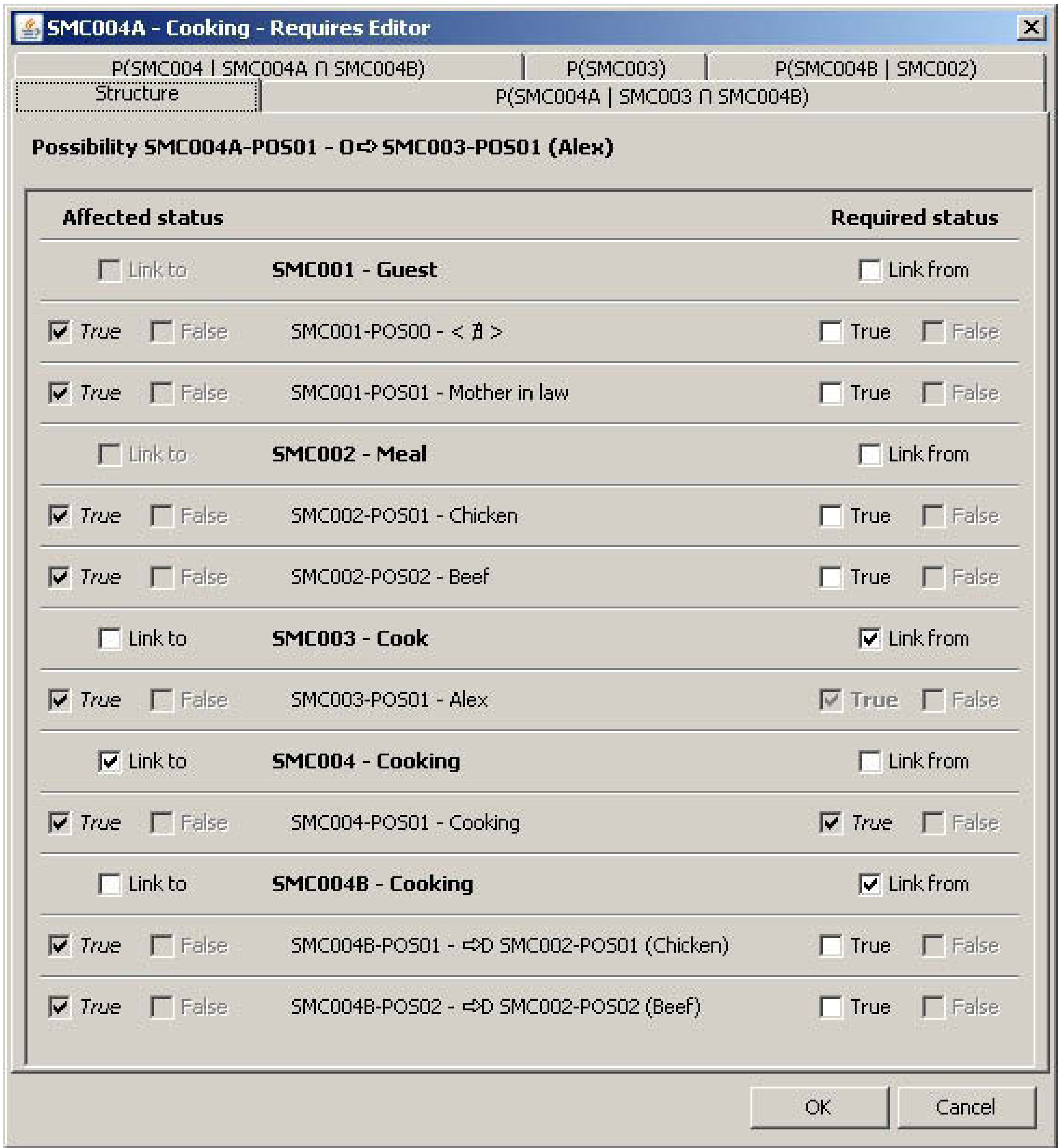



*Figure 36 Requires editor*

This panel shows the possibilities for which the requirements are to be added or removed, regrouped by component. It is possible to specify that the current possibility requires another possibility by setting the required status to “True” or “False” for the concerned possibility. If the current possibility affects another possibility, then the affected status column can be used.

If the checkbox label is enabled and bold, then the status has been explicitly defined by the user. If it is in italics, then it has been implicitly defined because of a transitive requires (see Section 3). However, if it’s bold but disabled, then it’s a system requires, such as a link end that inherently requires the target of the relation, since the link end cannot exists without its target. If the checkbox is unselected and disabled, then it means it is not allowed, because it would cause the possibility to be required true and false at the same time.

For each component of Figure 36, a checkbox allows to define probabilistic links between components, to be used by the Bayesian probabilities. With the Bayesian approach, the loopbacks are not allowed. Hence, if a non-direct path already exists between two components, it will not be possible to add a direct path in the other direction. In that case, the “Link to” or “Link from” checkbox will be disabled.

The main tab of the window of Figure 36 is the structure tab we just described, but other tabs exist to define the Bayesian probabilities. The probabilities for all the components of the current cluster are displayed here. Figure 37 shows the Bayesian Probabilities Tab.

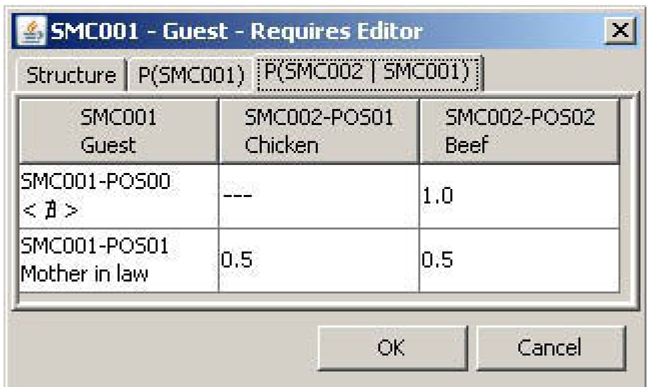



*Figure 37 Bayesian Probabilities Tab*

The Bayesian probabilities tab allows the user to edit the probabilities for each possibility, given the hypotheses defined by the links between the components. Each row must be equal to 1.0, and if the probability is ‘---‘, then it cannot be edited, because one of the possibilities requires the other to be false.

#### 4.3.2.2 Modifying an Element

The user can edit an existing situation element by clicking on the line corresponding to this element in the SM component view grid. This brings up the Element Properties panel (Figure 34). The “Apply” button is disabled until at least one modification has been done to the selected element. To delete an existing situation element, the user must click on the “Delete” button, and the component is removed from the situation.

### 4.3.3 Managing a Relation Component

To add a new relation component to the situation model, the user clicks on the “Add Relation” on the MHSA main frame (refer to Figure 33). This brings up the Element Properties panel shown in Figure 38.

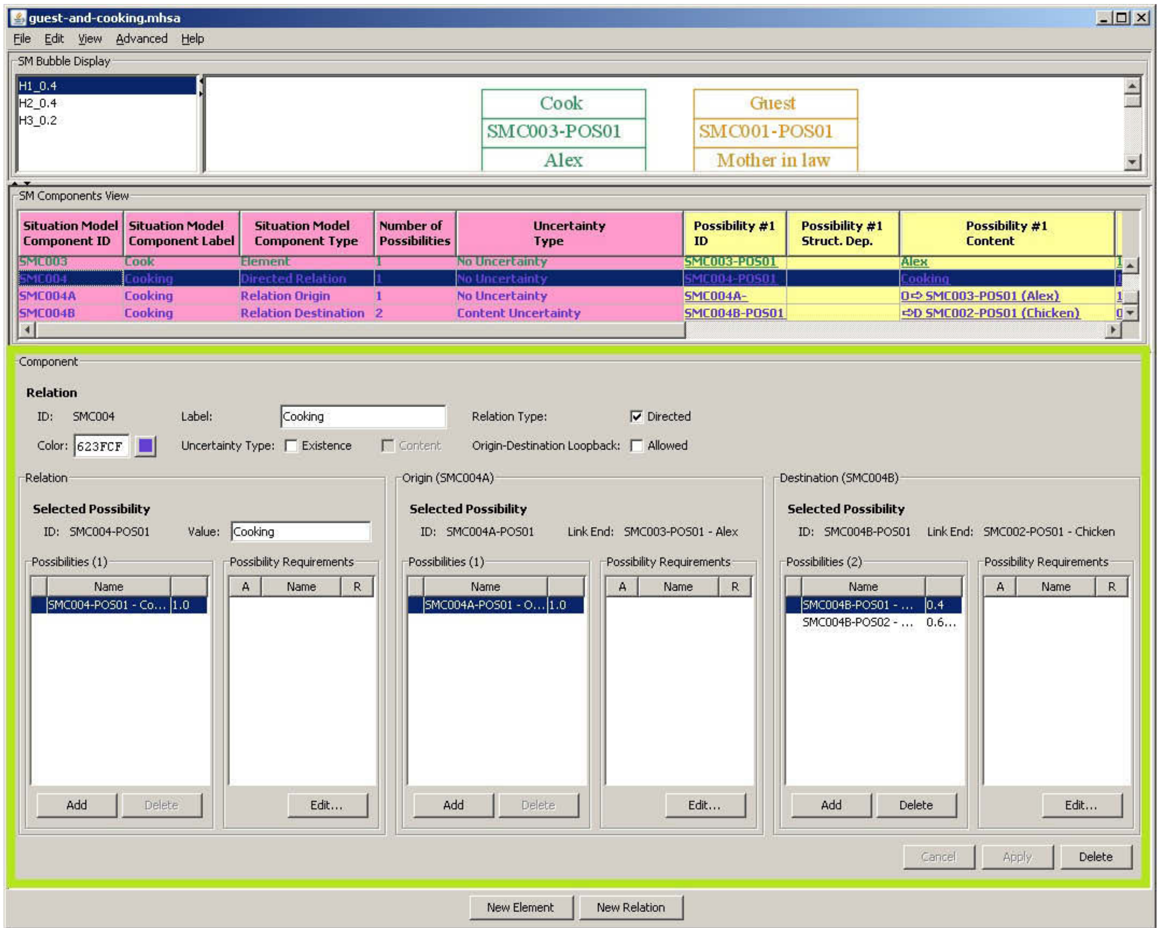



*Figure 38 Relation properties panel*

The Relation Properties panel is divided into three sections (from left to right on the figure): the Relation section, the Relation Origin section, and the Relation Destination section. The Relation Type indicates if the new relation is an undirected one or a directed one. If the user creates a directed relation this flag is checked, if the user creates an undirected relation this flag is unchecked. The Relation section works the same way as the “Element Properties” panel.

The Origin section allows the user to set the relation origin. To add an origin possibility, the user clicks on the “Add” button located under this list. This “Add” button brings up the Link End Selection panel (Figure 39).

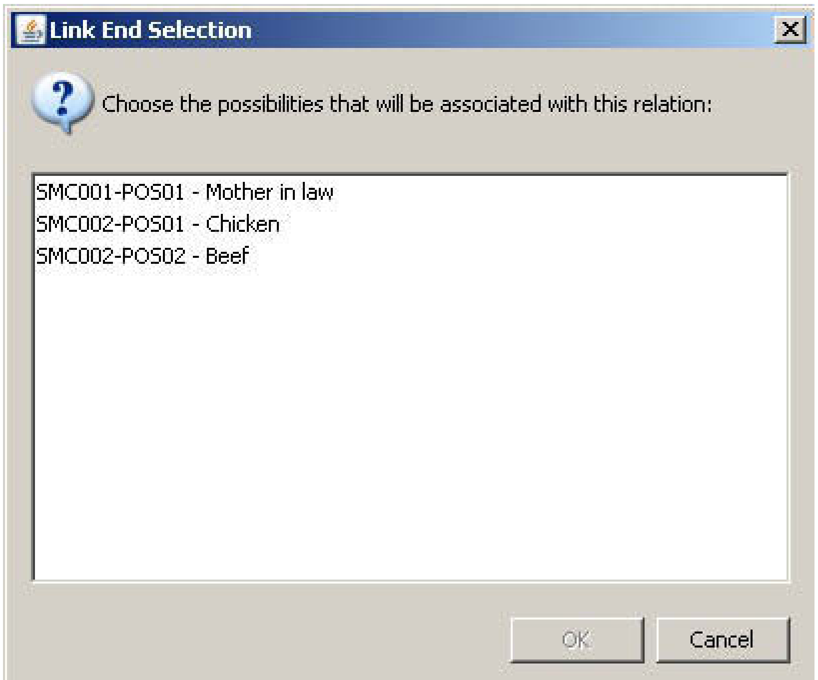



*Figure 39  Link end selection panel*

This panel allows the user to select one or more relation origin. A list shows all the possibilities. The “OK” button is disabled until the user makes a selection. Once the user selection is done, the user clicks on “OK” and the selections are added in the origin Possibilities list.

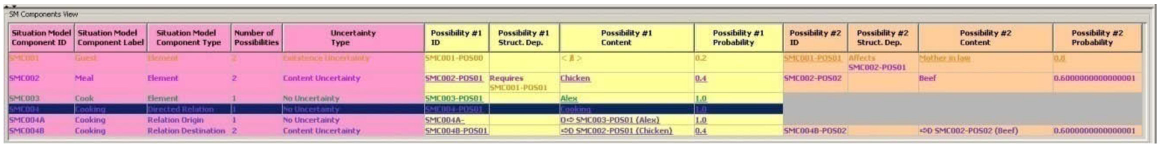
The Destination section allows the user to set the relation destination. The Destination section works that same way as the Origin section of the “Relation Properties” panel.

#### 4.3.3.1 Modifying a Relation

To edit an existing relation, the user clicks on the line corresponding to this relation in the SM components view grid. This brings up the Relation Properties panel (Figure 38). The user can modify the relation and, once satisfied with the changes, he/she clicks on the “Apply” button. To delete an existing situation relation, the user clicks on the “Delete” button, and the component is removed from the situation.

### 4.3.4 Situation Model Components View

The MHSASS prototype main frame has a panel showing all the components that have been added to the situation model. When the situation model changes, this panel is refreshed to always be synchronized with the current situation model. The user can select a component from this panel and the component properties are shown in the lower panel of the main frame. Figure 40 shows the SM Components View panel.



SM Components View

| Situation Model Component ID | Situation Model Component Label | Situation Model Component Type | Number of Possibilities | Uncertainty Type | Possibility #1 ID | Possibility #1 Struct. Dep. | Possibility #1 Content | Possibility #1 Probability | Possibility #2 ID | Possibility #2 Struct. Dep. | Possibility #2 Content | Possibility #2 Probability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SMC001 | Guest | Element | 2 | Existence Uncertainty | SMC001-POS00 | | < ∄ > | 0.2 | SMC001-POS01 | Affects SMC002-POS01 | Mother in law | 0.8 |
| SMC002 | Meal | Element | 2 | Content Uncertainty | SMC002-POS01 | Requires SMC001-POS01 | Chicken | 0.4 | SMC002-POS02 | | Beef | 0.6000000000000001 |
| SMC003 | Cook | Element | 1 | No Uncertainty | SMC003-POS01 | | Alex | 1.0 | | | | |
| SMC004 | Cooking | Directed Relation | 1 | No Uncertainty | SMC004-POS01 | | Cooking | 1.0 | | | | |
| SMC004A | Cooking | Relation Origin | 1 | No Uncertainty | SMC004A- | | O⇨ SMC003-POS01 (Alex) | 1.0 | | | | |
| SMC004B | Cooking | Relation Destination | 2 | Content Uncertainty | SMC004B-POS01 | | ⇨D SMC002-POS01 (Chicken) | 0.4 | SMC004B-POS02 | | ⇨D SMC002-POS02 (Beef) | 0.6000000000000001 |

*Figure 40 Situation model components view panel*

The SM component view shows all components (elements and relations) in detail. Every line of the grid contains a particular component with: its ID, its label, its type, its number of possibilities and its uncertainty type. If a particular component has numerous possibilities, they will all be shown on this panel. For a relation, the panel provides all details on three distinct lines for the relation itself, as well as its origin and destination.

### 4.3.5 Situation Model Bubble Display

The SM bubble display (Figure 41) shows the situation model corresponding to the hypothesis selected by the user.

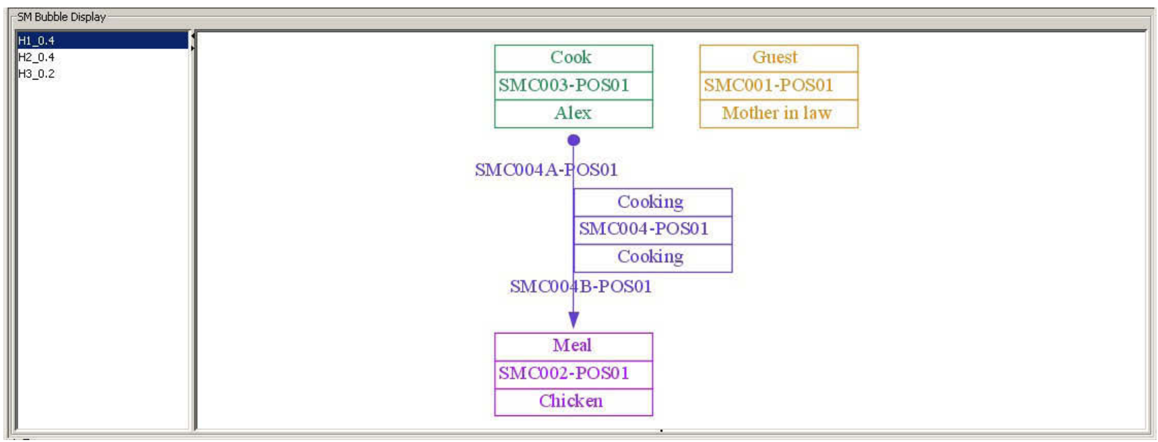



*Figure 41 Situation model bubble display*

The left window displays the hypotheses names and their likelihood scores. Once one of the hypotheses in selected within the left window, the corresponding situation model is displayed in the right window. The right window shows the model for the selected hypothesis using (square) bubbles, representing elements, and lines representing relations. If a relation is directed, it is shown as an arrow. Every element-bubble contains the element label (as defined by the user), unique ID, and the value of the represented possibility. The same information is shown for a relation in a square bubble adjacent to the line. In addition, for every relation, the IDs for the relation's origin and destination are shown at the beginning and end of the line.

### 4.3.6 Multiple Hypothesis Tree Display

It is possible to view the multiple hypothesis tree by going in the main menu bar, clicking on the "Advanced" menu, and choosing the "View MHTree" item. This brings up the Multiple Hypothesis Tree panel (Figure 42). This is an advanced option, not meant for regular users. It is intended to support deep analysis and understanding of the tree data structure by an advanced user (e.g., a system developer).

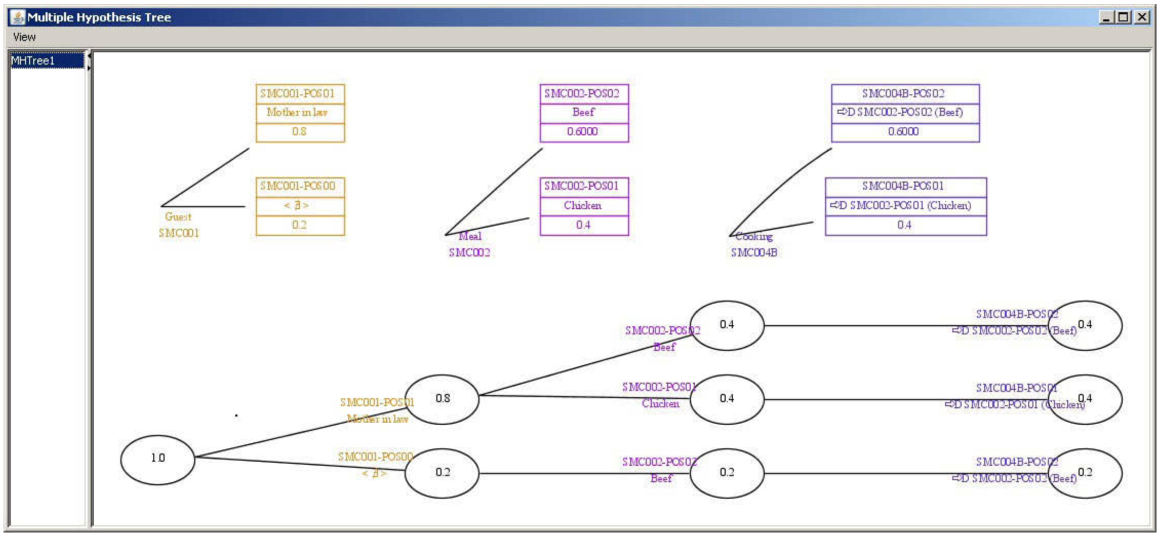



*Figure 42 Multiple Hypothesis Tree Panel*

If clustering is enabled (see 4.4.4), the user has the option to select one of the clusters in the left window. The selected cluster is shown in the right window. Every component of the situation model is shown at the top of the window with its corresponding possibilities label, ID and probability. The hypothesis tree is shown at the bottom of the window. Each possibility is displayed on a distinct branch with its ID and possibility label. The likelihood (probability in the present case) of a particular tree branch is displayed in the circle at the end of the branch. The probability displayed in a particular node (circle) reflects the cumulative chance of occurrence of a particular tree branch up unto that node. The probabilities shown in nodes at the (right) end of the tree (leaves), represent probabilities of hypotheses.

## 4.4 Configuring the MHSA Prototype

This section describes how to configure the various options for the MHSA prototype.

### 4.4.1 Configuring MHSA Systems Options

To change the MHSA parameters, the user goes in the main menu bar and clicks on the "Edit" menu and chooses the "Options" item. This brings the MHSA Properties panel.

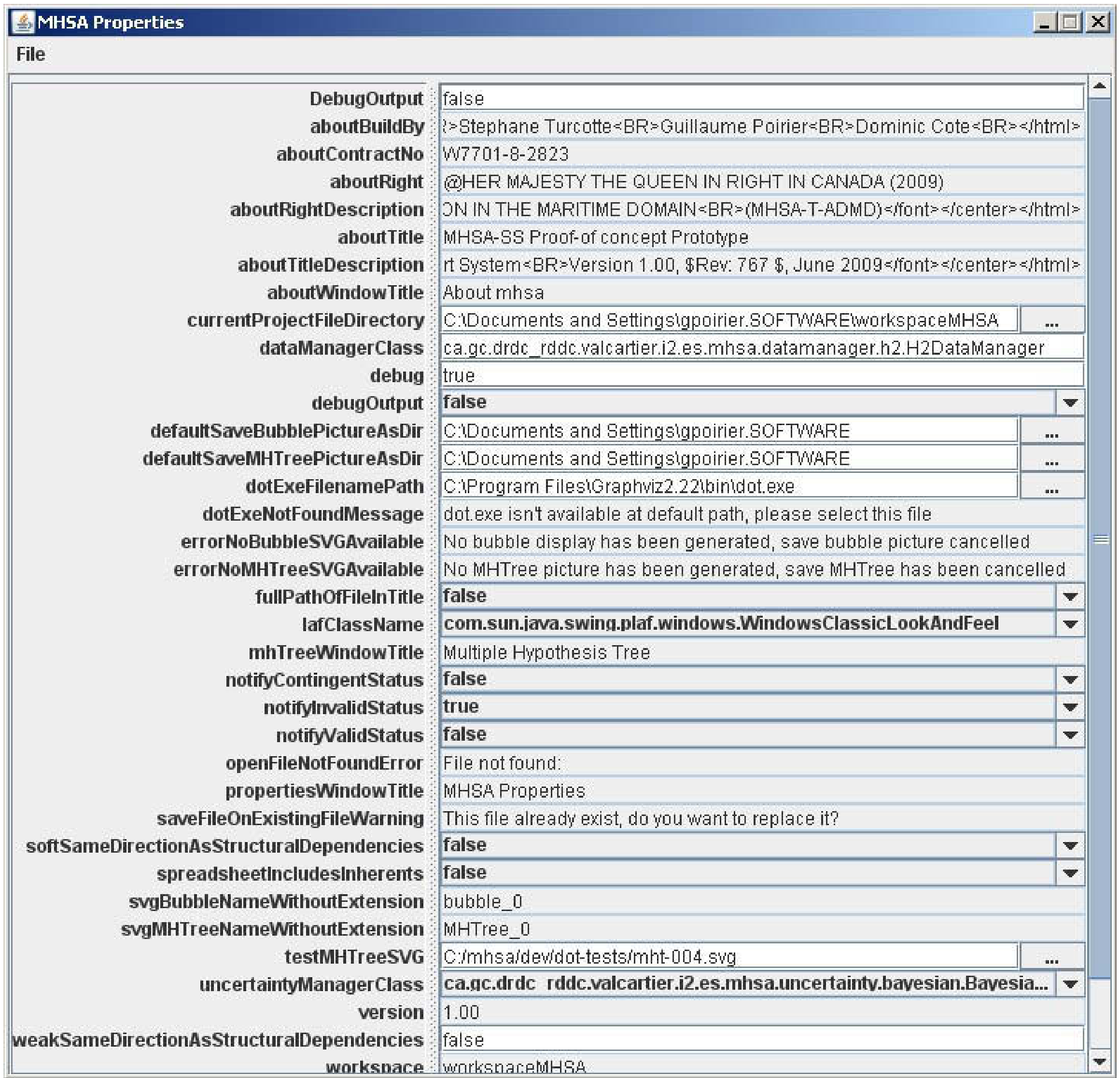



*Figure 43 MHSA properties panel*

Among other things, this panel allows the user to toggle the debug option, which gives more information on the software execution, and to specify default folders for screenshots and external libraries.

### 4.4.2 Configuring Display Properties

Display properties (Figure 44) are available through the Edit menu.

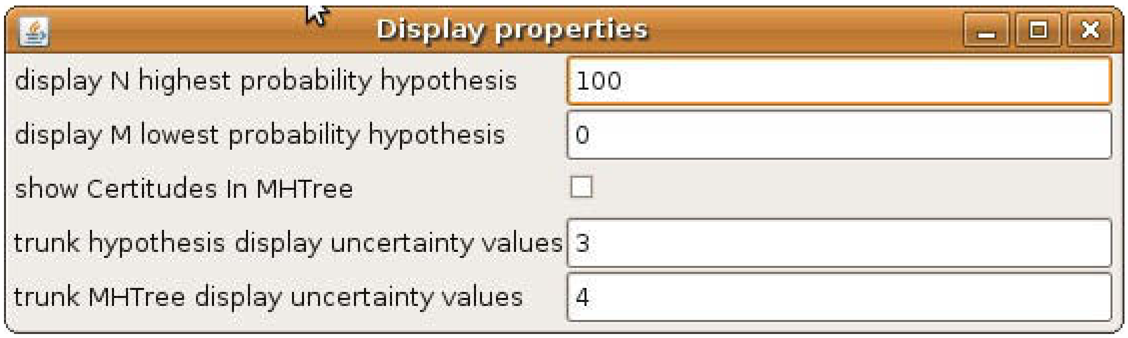



*Figure 44 Display properties panel*

Display configuration properties may be seen as three concepts, which are described next.

#### 4.4.2.1 Display a Subset of all the Hypotheses

The two parameters "display N highest probability hypothesis" and "display M lowest probability hypothesis" (defaulted at 100 and 0 respectively on Figure 44) allow the "filtering" of all the hypotheses generated to a specific sub-group on the Bubble display.

If both values are set to "0" (zero), then no filter is effective. The default setting will therefore only show the first 100 hypotheses.

#### 4.4.2.2 Display Clustered Certitudes When Displaying a Hypothesis Tree

By default, the "show Certitudes In MHTree" flag is unchecked as show on Figure 44. Activating this option shows additional levels in the MHTree for every "certain" component of the situation model. As components containing no uncertainty are common to every hypothesis, they are of limited interest when viewing the hypothesis tree, and they only contribute to additional cluttering.

#### 4.4.2.3 Uncertainty Value Display

The probability values of the listed hypotheses are displayed on the left panel of the Bubble display. Sometimes, they may be shown with too much precision (e.g., 0.33333333333333). To avoid cluttering that panel, a default length/precision is set to "3" (characters-truncation) via the parameter "trunk hypothesis display uncertainty values" as shown on Figure 44. The same goes for the probabilities shown in the Multiple Hypothesis Tree Display.

### 4.4.3 Configuring Hypothesis Management Properties

Hypothesis management properties (Figure 45) are available through the Edit menu.


*Figure 45 Hypothesis management properties*

Two pruning mechanisms have been implemented for the hypothesis tree and can be activated by selecting them with their respective check box. Pruning is a technique that reduces the size of the hypothesis tree by removing sections of the tree which are less relevant in order to reduce the size the tree data structure.

#### 4.4.3.1 Pruning Using Depth Prune Root Tree

Activating the “Depth Prune Root Tree Flag” will permit a maximum of N levels in the "uncertain section" of each hypothesis tree where N is the parameter “Depth Prune Root Tree At N levels” (proposed default value is 10 on Figure 45).

#### 4.4.3.2 Pruning Using Keep N Best Tree leaves

Activating the “Keep N Best Tree Leaves Flag” will permit a maximum of N leaves at the end of each hypothesis tree where N is the parameter “Keep N Best Tree Leaves Flag” (proposed default value is 100 on Figure 45).

Each pruning mechanism may be used with or without the other pruning mechanism.

### 4.4.4 Configuring Clustering

The Clustering flag (Figure 46) is available through the Edit menu.


*Figure 46 Clustering Flag Panel*

When the check box is checked as defaulted on Figure 46, clustering is enabled. Clustering allows the creation of distinct hypothesis trees for components that have no dependency between one another. When possible, having distinct trees limits the complexity and the size of the tree data structures and delays the potential use of pruning, which causes a loss of information.

When unchecked, all possibilities are processed under a single tree. Using MHSA with this option unchecked may lead to high memory usage for even relatively simple data sets. It is best to keep the option activated unless doing a simple small multiple hypothesis demonstration.

# 5 MHSA and Link Analysis

After completion of the MHSASS prototype, further investigation was required in order to assess existing MHSA capabilities in existing link analysis (LA) tools. A survey of existing link analysis tools was made, to gain a better understanding of these tools and how they deal with multiple hypothesis management [9]. Following this study, a specific LA tool was selected as a target for a study on the integration of the developed MHSA capability [9].

## 5.1 Survey of Link Analysis Tools

The survey of link analysis tools considered over forty LA/SA tools, which were analysed and documented. Particular focus was put on their respective capacity to execute multiple hypothesis management, which was globally found to be somewhat limited. Four specific tools (CoALA, I2 Analyst Notebook, Maltego and VisualLinks) were looked into in greater detail. The following criteria were used to compare the products: Data Analysis, Extraction, Categorization, Link Analysis, Visualization, Reporting, Deployment. CoALA was identified as the better choice for an eventual MHSA integration both because it offers the best features selection criteria, but also because it is not a “Foreign Ownership and Development”. DRDC-Valcartier CR 2009-322 [9] provides a detailed look at this survey.

## 5.2 Integration of MHSA into CoALA

Having identified CoALA as the best potential candidate for MHSA integration, a study was conducted to clearly identify the steps required to execute the integration. Anticipated modifications to both CoALA and the MHSA Prototype were presented, and the detailed set of components needed to realize the integration were presented. Key MHSA components to integrate were identified and documented. Finally, a global integration strategy was proposed. DRDC-Valcartier CR 2009-323 [9] provides a detailed look at the proposed integration process.

# 6 Conclusion

Throughout the previous sections, this document has provided a complete overview of MHSA. We have provided some theoretical look at MHSA, and discussed its governing principles. We have also provided detailed information on the MHSASS prototype developed under DRDC project 11hk. The prototype provides a good basis to understand and showcase the MHSA principles.

From a theoretical perspective, it would be of interest to revisit propositional logic's derived arguments to see how they could be rewritten and validated using xors instead of disjunctions. It may allow the construction of a complete mapping of the Situation Description Language and propositional logic's common operations. In turn, having such a mapping might allow managing Situation Description Language constructs with more common inference engines.

A full fledged MHSASS would prove a great asset to analysts trying to handle multiple hypotheses about uncertain situations, allowing them to defer decisions until new evidence comes in. It is currently planned to integrate a MHSA functionality into the Multi-Intelligence Tool Suite (MITS) developed at DRDC Valcartier. The MITS is a collection of integrated intelligence analysis tools. A MHSASS integrated into the MITS could leverage from automated reasoning and other Intelligence analysis functionalities, while providing a MHSA service.

# References

[1] Bosse, E., Roy, J. & Wark, S*., Concepts, Models, and Tools for Information Fusion*, Artech House Publishers, 2007.

[2] Endsley, M.R., *Toward a Theory of Situation Awareness in Dynamic Systems*, Jounal of the Human Factors and Ergonomics Society, Vol. 37, Number 1, March 1995, p. 32-64.

[3] Roy, J., *Toward Multiple Hypothesis Situation Analysis*, Proceedings of the 10th International Conference on Information Fusion (Fusion 2007), Paper 1236, Quebec, Canada, 9-12 July 2007.

[4] Roy, J., Bergeron Guyard, A., *Multiple Hypothesis Situation Analysis Support System Prototype*, 13th International Conference on Information Fusion (Fusion 2010), EICC, Edinburgh, UK, 26-29 July 2010.

[5] Chang, K. and Fung, R., *Target identification with Bayesian networks in a multiple hypothesis tracking system*, Opt. Eng. 36, 684 (1997).

[6] Roy, J., Duclos-Hindié, N., Dessureault, D., *Efficient cluster management algorithm for multiple hypothesis tracking*, SPIE 1997, 3163, 301 (1997).

[7] Reid, D.B., *An Algorithm for Tracking Multiple Targets*, IEEE Transactions on Automatic Control, Vol. AC-24, pp.843-854, December 1979.

[8] Mayrand, M., Duclos-Hindié, N., *Multiple Hypothesis Situation Analysis: CASE-ATTI Reusability Study*, Thales Systems Canada, DRDC Scientific Authority: Alexandre Bergeron Guyard, DRDC-Valcartier CR 2009-326, Defence R&D Canada - Valcartier, August 2009.

[9] Duclos-Hindié, N., *Multiple Hypothesis Situation Analysis: Review of Existing Link/Situation Analysis Tools*, Thales Systems Canada, DRDC Scientific Authority: Alexandre Bergeron Guyard, DRDC-Valcatier CR 2009-322, Defence R&D Canada - Valcartier, August 2009.

[10] Duclos-Hindié, N., *Multiple Hypothesis Situation Analysis: MHSA Integration with CoALA*, Thales Systems Canada, DRDC Scientific Authority: Alexandre Bergeron Guyard, DRDC-Valcatier CR 2009-323, Defence R&D Canada - Valcartier, August 2009.

[11] Wikipedia, *Propositional Calculus*, http://en.wikipedia.org/wiki/Propositional_logic.

[12] McCarthy, J., Hayes, P., *Some philosophical problems from the standpoint of artificial intelligence*. In B. Meltzer and D. Michie, editors, Machine Intelligence, 4:463–502. Edinburgh University Press, 1969.

This page intentionally left blank.

# Annex A MHSASS System Requirement Specification

This is a complete enumeration of the System Requirement Specification for the Multiple Hypothesis Situation Analysis (MHSA) Prototype.

## A.1 General Category, MHSA Requirements

| ID | Object Text | Category | Comments |
|---|---|---|---|
| **REQ-7** | Development should be executed using C++ or Java | General | C++ or Java are the most preferable languages for later integration. Any other choice should be at the approval of the scientific authority. |
| **REQ-151** | The Prototype should be runnable on Windows XP | General | Any other choice (ex. use of VMWare) should be at the approval of the scientific authority. |
| **REQ-152** | The Prototype shall be able to turn on and off the display of the hypothesis tree. | General | This requirement is mostly to be able to turn on and off the display of the hypothesis tree. |
| **REQ-153** | Prototype functionalities shall be controllable from a single program interface. | General | The prototype may use other peripheral/open-source tools but their control should be centralized in a single interface. |

*Table 10 General Category, MHSA Requirements*

## A.2 Performance Category, MHSA Requirements

| ID | Object Text | Category | Comments |
|---|---|---|---|
| **REQ-154** | Memory and performance will dictate the maximum number of hypothesis to be processed. | Performance | The goal here is to achieve the processing of a maximum number of hypotheses that memory can permit. If performances such as computing time are too affected, a maximum limit may be design or better yet, included as a configuration parameter defined by the user. |

*Table 11 Performance Category, MHSA Requirements*

## A.3 Usability Category, MHSA Requirements

| ID | Object Text | Category | Comments |
|---|---|---|---|
| **REQ-155** | The prototype shall run on a typical laptop | Usability | The word typical could include powerful laptops (RAM/disk space/CPU) in order to optimize performance requirement or to add features (such as VMware) that would compete for resources. |

*Table 12 Usability Category, MHSA Requirements*

## A.4 User Interaction Category, MHSA Requirements

| ID | Object Text | Category | Comments |
|---|---|---|---|
| **REQ-156** | Configuration shall be done before SA Model construction begins. | User Interaction | The user shall configure the system before the prototype execution. Ability to set that configuration from the main program interface would be preferable. |
| **REQ-157** | User shall be able to enter his SM components and their possibilities | User Interaction | The prototype via an interface shall capture the user situation model components. |
| **REQ-158** | The user shall be able to make corrections to situation model components and their possibilities. | User Interaction | In his/her situation design process, the user should be able to rename items, add and remove items. |
| **REQ-159** | Notification mechanisms shall warn the user that changes may impact other components | User Interaction | The difficult part would be to remove components as these objects will impact on other components which are dependent on them. (I.e. dependent probability changes, connected relations, etc). Depending on where was located the removed components, it could affect a small or large number of components. |
| **REQ-160** | The user shall be able to have an option for saving his/her inputs | User Interaction | The interface should allow the user to enter a file name for saving his/her situation model. |
| **REQ-161** | The user shall be able to enter and change the probabilities for his uncertain components. | User Interaction | This requirement will allow the user to experiment with his/her situation model and see how the final situation probabilities are affected. In particular, by transforming uncertain components into certain ones, the user should be able to view the collapse of the hypothesis |

| ID | Object Text | Category | Comments |
|---|---|---|---|
| | | | tree into a single situation. |
| **REQ-162** | The user shall be able to enter and change the model components' probability and the dependencies between his components. | User Interaction | |
| **REQ-163** | User defined probabilities shall be set between 0 and 1. | User Interaction | The user defines uncertainties according to the uncertainty model being used. Only the probability model is required to be implemented but the ability to addition of other models should be considered. |
| **REQ-164** | The system shall validate the probabilities (total equal to 1) and ensure that no dependencies cycles are allowed. | User Interaction | This requirement will depend on the uncertainty model used. In this case, the Bayesian model is assumed. |
| **REQ-165** | In the absence of values, the prototype shall assume equi-probability | User Interaction | For example, for 4 possibilities you will assign as a default value 0.25 for each. Again, this would be dependent of the uncertainty model used. |
| **REQ-166** | The prototype shall provide a central configuration utility. | User Interaction | Typically, either a configuration menu in a GUI or a configuration file editor from the prototype interface. Default values could be assigned. Configuration is assumed to be done before the creation of the hypothesis tree. |
| **REQ-167** | The prototype shall provide configuration for clustering. | User Interaction | An option for turning on/off the clustering should be available in the configuration menu or editor. |
| **REQ-168** | The prototype shall provide configuration for pruning. | User Interaction | An option for turning on/off the pruning capability should be available in the configuration menu or editor. Other parameters such as threshold or others should located at the same place. |

*Table 13 User Interaction Category, MHSA Requirements*

## A.5 Hypothesis Tree Management Category, MHSA Requirements

| ID | Object Text | Category | Comments |
|---|---|---|---|
| **REQ-169** | The management of the hypothesis tree shall be automatic | Hypothesis Tree Management | The creation of the hypothesis tree is automated based on the configuration parameters (clustering, pruning) and the user components and associated uncertainties. |
| **REQ-170** | A dynamic data structure shall be implemented for the hypothesis tree automatic capabilities. | Hypothesis Tree Management | This requirement is related to the reusability of the data structure used in CASE-ATTI for the hypothesis tree. |
| **REQ-171** | Hypothesis numbering and ranking shall be automatically managed. | Hypothesis Tree Management | The user would not have the possibility to modify a hypothesis number. The number should be somewhat natural for the hypothesis tree and not a random string of numbers. |
| **REQ-172** | The pruning capabilities of the prototype shall be automatic | Hypothesis Tree Management | The pruning of the hypothesis tree should only be based on the configuration parameters set before the creation of the tree and automated as the tree creation is started. |
| **REQ-173** | The clustering capabilities of the prototype shall be automatic. | Hypothesis Tree Management | The clustering of the hypothesis tree should only be based on the configuration parameters set before the creation of the tree and automated as the tree creation is started. |
| **REQ-174** | The prototype shall allow starting from an empty or previously saved situation hypotheses tree. | Hypothesis Tree Management | Database content is rebuild either from an empty restart or when loading previously saved situation hypotheses tree. |

*Table 14 Hypothesis Tree Management Category, MHSA Requirements*

## A.6 Data Management Category, MHSA Requirements

| ID | Object Text | Category | Comments |
|---|---|---|---|
| **REQ-175** | The user shall be able to save his/her partial or complete situation model and to retrieve it. | Data Management | The ability to save a situation model session is important in order to be able to retrieve it later to continue or reuse his/her situation analysis (ex: XML file, database, etc). |

| ID | Object Text | Category | Comments |
|---|---|---|---|
| **REQ-176** | The user shall be able to save the final hypothesis tree and to retrieve it. | Data Management | |
| **REQ-178** | The user shall be able to query the data for a given element. | Data Management | The user shall be able to examine data for any given component. |

*Table 15 Data Management Category, MHSA Requirements*

## A.7 Uncertainty Management Category, MHSA Requirements

| ID | Object Text | Category | Comments |
|---|---|---|---|
| **REQ-179** | The prototype shall support Bayesian uncertainty model. | Uncertainty Management | This requirement is a minimum required. Other uncertainty models can be supported. |
| **REQ-180** | Preferably, the uncertainty model shall be modular to allow the support of other uncertainty models in the future. | Uncertainty Management | Other possibilities in the future could include Dempster-Shafer, Fuzzy Logic or others. |
| **REQ-181** | The prototype shall calculate automatically the hypothesis score. | Uncertainty Management | Based on the uncertainty model used, the prototype should calculate and propagate along the hypothesis tree the uncertainty values. |
| **REQ-182** | The final hypothesis score shall be computed and displayed. | Uncertainty Management | Final probabilities of a given set of situations (selected/ranked by the user) are displayed with the associated possibilities for each situation. |
| **REQ-183** | A query mechanism shall allow for asserting uncertainty values to situation components for all situations. | Uncertainty Management | |
| **REQ-184** | A query mechanism shall allow for asserting uncertainty values to Hypotheses | Uncertainty Management | The MHSA Prototype shall compute/provide a mechanism to assess uncertainty values to Hypotheses, which the user shall be able to read/query. |

*Table 16 Uncertainty Management Category, MHSA Requirements*

## A.8 Visualization – Situation Models Category, MHSA Requirements

| ID | Object Text | Category | Comments |
|---|---|---|---|
| **REQ-185** | Visualization means shall be provided to present the situation models and their individual final probability graphically. | Visualization - Situation Models | The display should include the final probabilities, and its associated situation characteristics, using a graph model diagram |
| **REQ-186** | The user shall be able to query for specific components probability | Visualization - Situation Models | For example, to query the probability of the situations containing the use of hardkill (in opposition to softkill) response. |
| **REQ-188** | The prototype shall provide situation model ranking/sorting capability | Visualization - Situation Models | The prototype shall provide a ranking/sorting capability based on: numbering, probability, subset (N top hypothesis, probability > threshold (10%), any hypothesis containing a given possibility for an element) |

*Table 17 Visualization – Situation Models Category, MHSA Requirements*

## A.9 Visualization – Hypothesis Tree Category, MHSA Requirements

| ID | Object Text | Category | Comments |
|---|---|---|---|
| **REQ-187** | The prototype shall allow the optional display of the hypothesis tree. | Visualization - Hypothesis Tree | This can be achieved either by a configuration option, a user/developer mode setup or a display button in the interface. This Hypothesis tree visualization is only aimed at helping "scientists"/"developers" examine MHSA internal processing. |

*Table 18 Visualization – Hypothesis Tree Category, MHSA Requirements*

# List of symbols/abbreviations/acronyms/initialisms

| | |
|---|---|
| ARP | Applied Research Project |
| CASE-ATTI | Concept Analysis and Simulation Environment for Automatic Target Tracking and Identification |
| COA | Course of Action |
| LA | Link Analysis |
| MHSA | Multiple Hypothesis Situation Analysis |
| MHSASS | Multiple Hypothesis Situation Analysis Support System |
| MHTree | Multiple Hypothesis Tree |
| MHT | Multiple Hypothesis Tracking |
| PL | Propositional Logic |
| RAM | Requires/Affects Manager |
| SA | Situation Analysis |
| SDL | Situation Description Language |
| SM | Situation Model |
| SMC | Situation Model Component |
| SRS | System Requirement Specification |
| VDQ | Ville de Québec |

This page intentionally left blank.

## DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

   Defence R&D Canada – Valcartier
   2459 Pie-XI Blvd North
   Quebec (Quebec)
   G3J 1X5 Canada

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

   UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

   Multiple Hypothesis Situation Analysis

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

   Bergeron Guyard, A.; Roy, J.

5. DATE OF PUBLICATION (Month and year of publication of document.)

   February 2011

6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

   84

6b. NO. OF REFS (Total cited in document.)

   8

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

   Technical Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

   Defence R&D Canada – Valcartier
   2459 Pie-XI Blvd North
   Quebec (Quebec)
   G3J 1X5 Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

   11hk

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

   W7701-W7701-082823/001/QCL

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

   DRDC Valcartier TR 2010-525

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

   Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

   Unlimited

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

In 2008, DRDC undertook Applied Research Project (ARP) 11hk entitled “Multiple Hypothesis Link Analysis for Anomaly Detection in the Maritime Domain” (MHLA-4-ADMD). The main objective of this ARP is to study, develop and implement a situation analysis tool to support anomaly detection, the identification of vessels of interest (VOI), and threat analysis in the maritime domain. In this context, it was decided to explore the concept of Multiple Hypothesis Situation Analysis (MHSA) and to develop a MHSA Support System (MHSASS) prototype. This document is meant to provide a complete overview of MHSA. The objective is twofold. Firstly, provide a theoretical look at MHSA, and discuss its governing principles. Secondly, provide detailed information on the MHSASS prototype that was developed. This document explains the concepts driving MHSA. It shows how MHSA works and why it could be of great use to the operators. This work constitutes a milestone that could lead to the creation of a system that would help analysts deal with the uncertainty that makes (even simple) situation difficult to analyse.

En 2008, RDDC a entrepris un projet de recherche appliqué (PRA) intitulé "*Multiple Hypothesis Link Analysis for Anomaly Detection in the Maritime Domain*" (MHLA-4-ADMD). L'objectif principal de ce PRA est d'étudier, de développer et d'implémenter un outil d'analyse de la situation pour supporter la détection d'anomalies, l'identification de *vessels of interest* (VOI) et l'analyse de la menace dans le domaine maritime. Dans ce contexte, il fut décidé d'explorer le concept de *Multiple Hypothesis Situation Analysis* (MHSA) et de développer un prototype de MHSA *Support System* (MHSASS). Ce document a pour but de fournir un survol complet de MHSA. Un objectif est de donner un aperçu théorique de MHSA, et de ses principe gouvernants. Un second objectif est de fournir de l'information détaillée sur le prototype MHSASS qui a été développé. Ce document explique les concepts qui guident le MHSA. Il montre comment MHSA fonctionne et pourquoi il pourrait être d'une grande utilité pour les opérateurs. Ce travail constitue un point tournant qui pourrait mener à la création d'un système qui pourrait aide les analystes à gérer l'incertitude qui rend les situations (même les plus simples) difficiles à analyser.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Multiple Hypothesis; Situation Analysis; Sensemaking; Link Analysis

**Defence R&D Canada**

Canada's Leader in Defence
and National Security
Science and Technology

**R & D pour la défense Canada**

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE R&D DÉFENSE

**www.drdc-rddc.gc.ca**