

Multiple Hypothesis Situation Analysis Support System Prototype

Jean Roy

Intelligence and Information Section
Defence R&D Canada – Valcartier
Quebec, Qc, Canada

Jean.Roy@drdc-rddc.gc.ca

Alexandre Bergeron Guyard

Intelligence and Information Section
Defence R&D Canada – Valcartier
Quebec, Qc, Canada

Alexandre.Bergeron-Guyard@drdc-rddc.gc.ca

Abstract - Uncertainty makes the analysis of even simple situations difficult. It forces intelligence analysts to formulate and manage hypotheses during the construction of explicit representations of real world situations. This may quickly become overwhelming. To provide better support to the intelligence staff, the main concepts behind multiple hypothesis tracking have been revisited to develop a proof-of-concept prototype of a multiple hypothesis situation analysis (MHSA) support system. A key objective is to showcase the potential and utility of MHSA. It has thus been conceived to allow users, developers, and managers to better understand each and every aspect of the MHSA process, which isn't like a Bayesian Net. This paper discusses a situation modeling graphical language, the interdependency and uncertainty about the situation model components, the hypothesis tree data structure used to keep track of the uncertainty, different issues regarding the hypothesis tree, hypothesis scoring, and user interactions with the MHSA support system prototype.

Keywords: Situation modeling, situation analysis, uncertainty, multiple hypotheses, hypothesis tree.

1 Introduction

Uncertainty forces intelligence analysts to formulate and manage hypotheses during the construction of explicit representations of real world situations. Because of human cognitive limitations, this may quickly become overwhelming, even for the most experienced and capable analysts. To provide better support to the intelligence staff dealing with uncertainty in situation analysis, the main concepts behind Multiple Hypothesis Tracking (MHT) have been revisited to develop a proof-of-concept prototype of a Multiple Hypothesis Situation Analysis Support System (MHSA-SS) [1]. A key objective with this prototype is to showcase the potential and utility of MHSA. It has thus been conceived to allow users, developers, and managers to better understand all aspects of the MHSA process, which isn't like a Bayesian Net.

This paper first presents the graphical language made available to the analysts to create representations or models of the situations under examination. In the MHSA

framework, there is uncertainty when there are more than one mutually exclusive possibilities for the existence and/or the contents of any given situation model component. Drawing from the MHT approach, a hypothesis tree data structure is used to keep track of this uncertainty and of the corresponding multiple situation models that must be maintained in parallel. The paper discusses different issues regarding this hypothesis tree, with emphasis given to the identification and management of the explicit and implicit dependencies that arise from the relationships between the situation model components. Hypothesis scoring is also discussed, with examples provided for the probability framework initially implemented for the prototype. The user interactions with the support system are described, along with the displays used to visualize the situation models built by the user and the corresponding hypothesis tree(s). The paper concludes with potential future work related to MHSA.

2 Situations and situation analysis

Situation analysis (SA) has previously been defined as “a process, the examination of a situation, its elements, and their relations, to provide and maintain a product, i.e., a state of situation awareness, for the decision maker” [2]. The SA process encapsulates that part of the overall decision-making cycle that is concerned with understanding the world. There is a real situation unfolding in the environment, and the SA process will create and maintain a representation of it.

The purpose of a computer-based situation analysis support system (SASS) is to assemble an explicit representation (i.e., a model) of aspects of interest in an environment. This situation model, that the SA process endeavours to keep up to date, is not only a representation of the various elements of the situation, but also a representation of how they relate.

A situation can be defined as a specific combination of circumstances, i.e., conditions, facts, or states of affairs, at a given moment. In line with this, one can say that a situation is a combination of situation components. Some basic situation components that are relevant to most military and public security operations include entities, identity, kinematics, sensors, weapons, capabilities, intentions, behaviours, actions, impacts, threats, terrain,

etc. In the context of information fusion and situation analysis systems, the situation components are typically characterized as being either ground truth, observed, estimated, fused, inferred, smoothed, filtered, and/or predicted/projected. A simple situation is illustrated in Fig. 1, here graphically presented as it would be on the display of a typical link analysis tool.

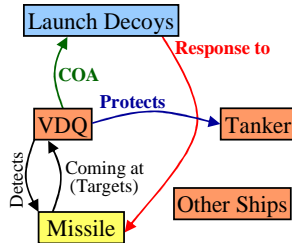


Figure 1: A simple situation example

The components of this situation are as follows:

- There is a frigate, the *Ville de Québec* (VEQ), a tanker, and some other ships in the area of interest of the analyst.
- The VEQ's mission is to protect the tanker.
- The analyst is informed that VEQ's radar has provided a contact on a missile.
- He/she infers the missile is targeting the VEQ.
- The analyst projects that the VEQ will launch decoys to protect itself (and the tanker).

Figure 1 clearly shows the individual situation elements, and the relationships between these elements. Note also that this example contains observed, inferred, and projected situation components.

3 Graphical situation modeling language

A graphical language is required for the construction of explicit representations of real world situations, like the one shown in Fig. 1. Such a language, different from the one used for Bayesian Networks, has been defined.

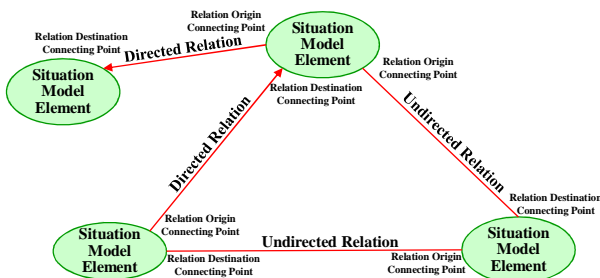


Figure 2: Graphical situation modeling language

The language shown in Fig. 2 allows the intelligence analysts to define and manipulate *situation model components* (SMCs) to create graphical representations of situations. The language is limited to five types of SMCs that can be used by the analysts: 1) *Element*, 2) *Undirected Relation*, 3) *Directed Relation*, 4) *Relation Origin Connecting Point*, and 5) *Relation Destination*

Connecting Point. Everything that an analyst has to say about a given situation must be expressed using only these five types of SMCs.

One should note that only one SMC is required to define a situation element, while three SMCs are required to define a relationship (i.e., the relation itself, its origin, and its destination).

4 Situation model uncertainty

In the presence of uncertainty, the analysis of the simple situation shown in Fig. 1 can quickly become more complex. For example, what if the origin of the contact provided by the source is uncertain? What if the analyst doesn't know for sure if the contact is a missile or a false alarm (resulting from sensor limitations)? If the contact is believed to be a missile, what if there is uncertainty in the evidence used to infer that the VEQ is the target of this missile? Because of this uncertainty, the analyst may consider that the protected tanker is the actual target, or it could also be one of the other ships in the vicinity. Finally, there might be different courses of action (COA) that the commander of the VEQ could use to defend itself or the tanker. The commander may also decide to do nothing if he/she believes that some other ship is the actual target. Hence, while generating the projection of the situation, the analyst doesn't know with certainty which COA will be selected and implemented.

The graphical situation modeling language introduced in Section 3 must allow the analysts to express the kind of uncertainty described above.

4.1 Certainties and possibilities

Within the proposed MHSA framework, one says that there is uncertainty when there are more than one possibility for a given situation model component. Moreover, when there are multiple possibilities for a situation model component, these possibilities *must be mutually exclusive*; if one is true, then all the others are necessarily false. When there is only one possibility for a situation model component, then this possibility is considered as certain.

4.2 Existence and content uncertainty

Within the MHSA framework, there are only three types of uncertainty that can be associated to a situation model component: 1) existence, 2) content, and 3) existence and content. A given SMC may be considered to exist or not in the situation being modeled. When a SMC is considered to exist, then the second type of uncertainty has to do with the contents of this component, when there are multiple possibilities for these contents. The third type is a combination of the other two: one is not certain that a given component exists, and he/she may also not be sure about its contents if the component exists. Whatever the uncertainty type, *it will always be expressed as different possibilities for the situation model component*.

In this framework, the existence of a SMC of type *relation* always requires: a) its intrinsic existence, b) the existence of the situation element from which it originates (i.e., the *Relation Origin Connecting Point*), and c) the existence of the situation element being pointed to (i.e., the *Relation Destination Connecting Point*). Hence, if the situation element at the origin (or destination) of a relation ceases to exist, then this relation must also cease to exist.

5 Multiple situation hypotheses

Facing the uncertainty mentioned above, the analyst will necessarily have to formulate hypotheses regarding the situation. One such hypothesis could be:

- the contact is from a missile,
- the missile is targeting the tanker, and,
- the VDQ will use a hard-kill weapon.

Another hypothesis that is similar, but nevertheless distinct, would be that:

- the contact is from a missile,
- the missile is targeting the tanker, and,
- the VDQ will launch decoys.

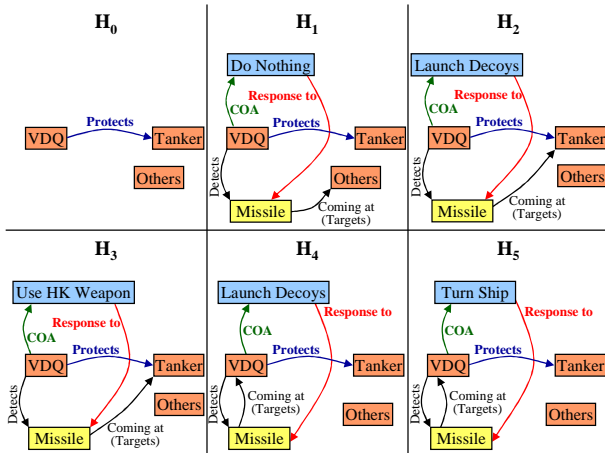


Figure 3: Multiple situation hypotheses

Figure 3 shows, in a graphical form, six such hypotheses that could be formulated by the analyst. Note that in hypothesis H₀ the contact is considered to be a false alarm, and nothing else happens (Figs. 3 and 4). There are a few things to note about these multiple hypotheses:

- The elements of the situation that are known with certainty are present in all hypotheses. Such elements are the existence of the entities “VDQ”, “tanker”, and “other ships”, and also the fact that the “VDQ protects the tanker” (i.e., a known relationship).
- It is presumed that one of the hypotheses actually corresponds to the true situation.
- The analyst could either decide immediately on which hypothesis is the correct one, running the

risk of being wrong, or defer the decision until more evidence confirms one hypothesis.

Maintaining multiple situation hypotheses actually corresponds to maintaining multiple situation models (or multiple “possible worlds”) in parallel within the support system database; each model is represented as a set of interconnected situation elements.

6 Hypothesis tree representation

Because of cognitive limitations, the analyst can quickly lose track of all of the possibilities. There is thus a need to organize the hypotheses in such a way that it becomes easier to visualize and manage them. In this regard, the hypotheses can be organized using a tree-like, graphical data structure. Figure 4 illustrates such a hypothesis tree, corresponding to the hypotheses of Fig. 3.

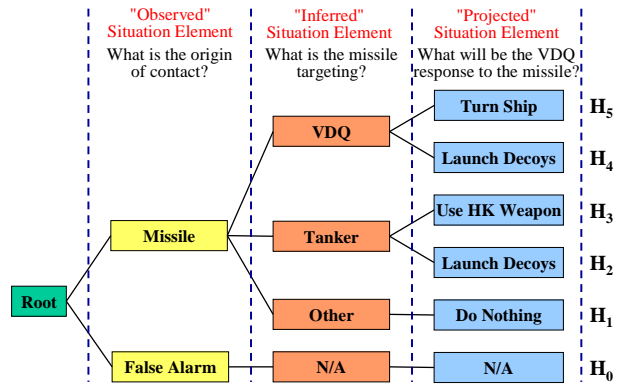


Figure 4: Hypothesis tree representation

6.1 Equivalence of representations

A key aspect is that the hypothesis tree graphical representation (Fig. 4) of the possible situation *must be equivalent* to the “bubbles and links” graphical representation (Fig. 3). 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 (Figs. 3 and 4) can be entirely constructed without one having to care about probabilities at all.

6.2 Containers without semantics

Another very important aspect is that the situation model components of types *element* and *relation* in Fig. 2 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 MHS 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 Fig. 5.

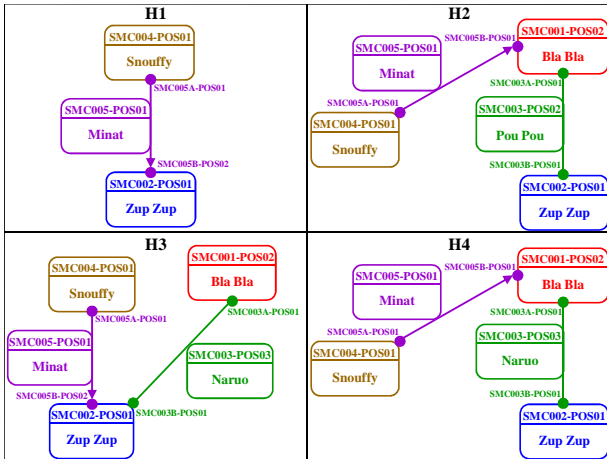


Figure 5: Containers with meaningless contents

There is certainly a semantics related to the graphical language itself. For example, the MHS-SS 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 MHS-SS 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 6 shows the hypothesis tree matching the hypothetical part of the situation modeled in Fig. 5. Note that SMC002, SMC003A, SMC003B, SMC004, SMC005 and SMC005A are all *certain* components in Fig. 5. As such, they are present in all hypotheses of Figs. 5 and 6.

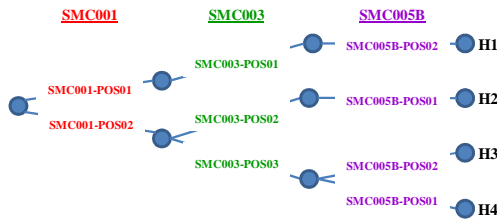


Figure 6: Hypothesis tree with meaningless containers

6.3 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.

7 Dependency of component possibilities

During the management of the “hypothesis tree” and “bubbles and links” graphical representations, one has to care about the dependencies between the possibilities of the different SMCs. As such dependencies are reflected into the structure of the hypothesis tree, they are called *structural dependencies*. If a decision made on the possibilities for a given SMC has any impact on the possible decisions about the possibilities of another SMC, then these two components are *structurally dependent*.

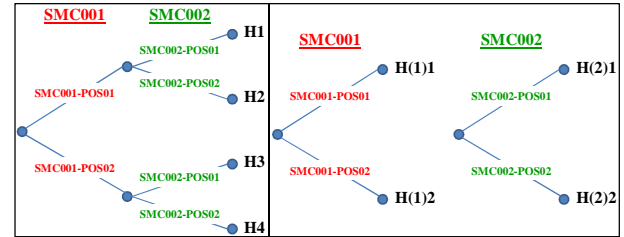


Figure 7: Structurally independent SMCs

An example where two situation model components (*SMC001* and *SMC002*) are *structurally independent* is shown in Fig. 7. In this example, if a decision is made for *SMC002* to keep only the possibility *SMC002-POS02* (for example), then the candidate decisions for *SMC001* are not affected; *SMC001* can still be *SMC001-POS01* or *SMC001-POS02*. Similarly, if a decision is made for *SMC002* to keep only the possibility *SMC002-POS01*, then the candidate decisions for *SMC001* are not affected; *SMC001* can still be *SMC001-POS01* or *SMC001-POS02*. One could also consider a decision on the possibilities for *SMC001* with the same kind of conclusions.

As shown on the right-hand side of Fig. 7, SMCs that are *structurally independent* can be represented in separate hypothesis trees. That is, the hypothesis tree on the left-hand side of Fig. 7 could (and should) be split into two hypothesis trees. Actually, this is the essence of hypothesis clustering, a technique used to reduce the amount of hypotheses that must be maintained in any given hypothesis tree. When a large tree is split into smaller trees, each overall situation hypothesis is a combination of one hypothesis from each of the sub-trees. In Fig. 7 for example, hypothesis *H3* is the combination of hypothesis #2 of sub-tree #1 and hypothesis #1 of sub-tree #2. That is, $H3 = H(1)2$ and $H(2)1$.

An example where two situation model components are *structurally dependent* is shown in Fig. 8. In this example, if a decision is made for *SMC002* to keep only the possibility *SMC002-POS03*, then the only candidate decision for *SMC001* becomes necessarily *SMC001-POS02*; it is impossible to have a situation with *SMC001-POS01* if *SMC002-POS03* is selected as the only option for *SMC002*. Similarly, if a decision is made for *SMC001* to keep only the possibility *SMC001-POS01*, then the possibility for *SMC002* to be *SMC002-POS03* is necessarily eliminated.

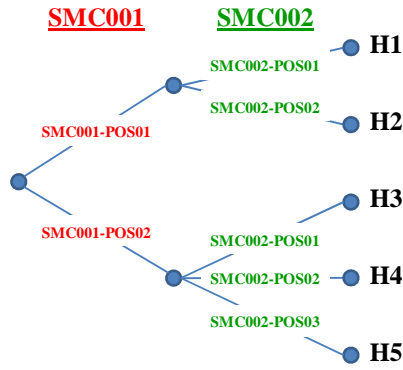


Figure 8: Structurally dependent SMCs

7.1 Sources of structural dependencies

Structural dependencies may arise for various reasons. For example, pruning branches of the hypothesis tree for growth management purpose may introduce « artificial » dependencies between components. In the left-hand side of Fig. 7, pruning hypothesis *H3* to keep only 3 hypotheses would automatically create a structural dependency between *SMC001* and *SMC002*.

Structural dependencies may also be desirable from a situation modeling perspective. Hence, the MHSA-SS prototype provides the user with a functionality to define such dependencies. That is, when a possibility is created for one component, it can be made dependent on the possibilities of other components.

7.2 « Requires » type of dependencies

Only two types of interdependency for component possibilities are allowed in the current prototype: *Requires True*, and *Requires False*. Here the words *True* and *False* refer to the presence (or non-presence) in the hypothesis tree of a possibility seen as a container: these words don't refer to the truth value of the actual contents of the possibility (the MHSA-SS understands and manipulates containers, not their contents).

A possibility may be said to *require another possibility to be true* for itself be allowed to be true. That is, if the *required true* possibility is true, the requiring possibility may be either true or false. However, if the *required true* possibility is not true (i.e., is not present), then the requiring possibility is necessarily not valid (i.e., not present). These cases are shown in Table 1.

Table 1: Cases for « *A Requires True B* »

Possibility A	Possibility B	<i>A Requires True B</i>
False	True	<i>A can be false when B is true</i> <i>B can be true when A is false</i>
False	False	<i>A must be false when B is false</i> <i>B can be false when A is false</i>
True	True	<i>A can be true when B is true</i> <i>B must be true when A is true</i>

A possibility may be said to *require another possibility to be false* for itself be allowed to be true. That is, if the *required false* possibility is false, the requiring

possibility may be either true or false. However, if the *required false* possibility is true (i.e., is present), the requiring possibility is necessarily not valid (hence false, or not present). These cases are shown in Table 2.

Table 2: Cases for « *A Requires False B* »

Possibility A	Possibility B	<i>A Requires False B</i>
False	True	<i>A must be false when B is true</i> <i>B can be true when A is false</i>
False	False	<i>A can be false when B is false</i> <i>B can be false when A is false</i>
True	False	<i>A can be true when B is false</i> <i>B must be false when A is true</i>

7.3 Taking into account dependencies

The information on structural dependencies is essential to the MHSA process. It is used: 1) during the expansion of the hypothesis tree, when processing a new SMC, to determine if branching from an existing node with a given possibility of the new component is allowed, and 2) to manage the tree when a SMC is removed (with all of its possibilities), or more simply when a given possibility for a given SMC is removed.

As illustrated in Fig. 9, the concept of a component possibility (say possibility *a* of component *X*) *requiring true* a possibility of another component (say possibility *b* of component *Y*) provides a means to specify if one can or not create a new branch during the expansion of the hypothesis tree. That is, if such a “require” statement is in force, then a new branch associating possibility *a* to component *X* can be joined to an existing tree node only if possibility *b* is associated with component *Y* (according to this existing node). All hypotheses associating component *X* with *a* must also associate component *Y* with *b*.

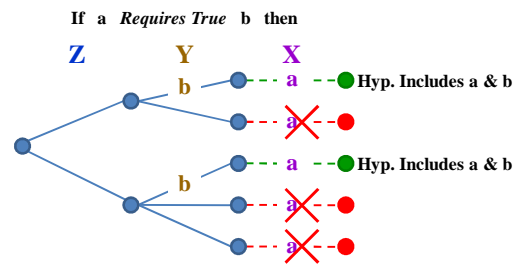


Figure 9: Dependencies and tree expansion

Since possibilities are assumed to be mutually exclusive, *associating* a component to one of its possibilities in a given hypothesis implies *not associating* any of the “competing” possibilities (of that same component) to this component. Hence, each hypothesis assumes 1) a single possibility to be *true* (associated) for a given component, and 2) all other competing possibilities to be *false* (not associated) for that same component. Using such a *true/false* semantics allows the *requires specifications* to specify that a component *must be associated* (or that it *must not be associated*) with one of its possibilities for another possibility of another component to be associated with this other component.

7.4 « Affects » type of dependencies

If a possibility for a component can be made dependent on the possibilities of other components through some «requires» relationships, then the reciprocal «affects» relationships must also be considered. Hence, possibility B is said to *true affect* possibility A if possibility A *requires true* possibility B . The cases for B *true affect* A are identical to those shown in Table 1 for A *requires true* B . Similarly, when possibility A *requires false* possibility B , then possibility B is said to *false affect* possibility A . The cases for B *false affect* A are identical to those of Table 2.

7.5 Analogy with the logical implication

A useful parallel can be drawn between the “requires” dependencies in the MHSA framework and the logical implication/entailment in logics. Table 3 provides the truth table of the logical implication.

Table 3: Truth table of the logical implication

A	B	$A \rightarrow B$
False	True	True
False	False	True
True	True	True
True	False	False

The implication means that when $A \rightarrow B$ is true, then A can only be true if B is true; however, A is not necessarily true when B is. It also means that A is necessarily false when B is false. This is very similar to the “requires true” concept for MHSA (as described in Table 1). The utility of this analogy is that it inspires the application of typical logical reasoning to the management of the structural dependencies between the SMCs. Reference 4 documents all of the logical rules that have been identified and used by the *Requires/Affects Manager* of the MHSA-SS prototype to support the enforcement of the *requires/affects* structural dependencies during the management of the hypothesis tree.

7.6 Explicit and implicit dependencies

A key aspect of the *Requires/Affects Manager* is that it uses the logical rules to automatically deduce all the implicit dependencies resulting from the explicit dependencies defined by the user or from the management of the tree. For example, adopting the notation $A \rightarrow B$ to mean A *requires true* B , then one can define the *implicitly requires true* relationship. That is, if possibility A *requires true* possibility B which in turn *requires true* possibility C , then possibility A is said to *implicitly requires true* possibility C . Using the compact notation, if $A \rightarrow B$ and $B \rightarrow C$, then $A \rightarrow C$. Note also 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. Finally, note that an “explicit” *requires true* may be considered also “implicit”, but not the reverse.

A more subtle case arises from the assumption that the possibilities for a given component must be mutually exclusive. If a possibility A *requires true* (or *implicitly requires true*) a possibility B for a given component, then possibility A is said to *implicitly require false* all other possibilities (C, D, E , etc.) of this other component. Many other subtle cases are documented in Ref. 4, including those arising from hypothesis pruning.

8 Hypothesis scoring and uncertainty

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.

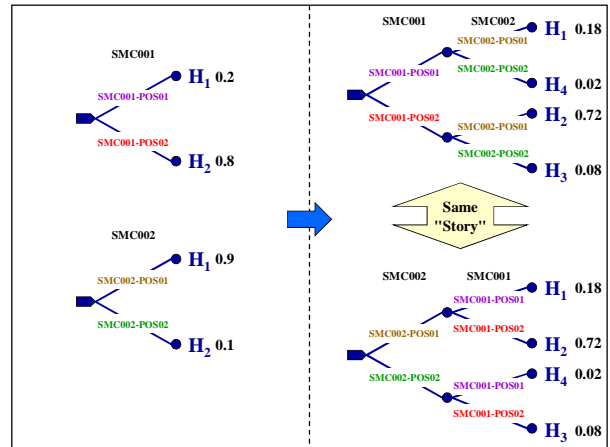


Figure 10: Computing probabilities – independent case

In particular, the notion of structural dependency between SMCs in the MHSA framework must not be mixed with the notion of conditional dependency in the domain of probabilities. Although these two notions can sometime be related in some scenarios, the two concepts are not the same. Figure 10 shows a simple example where two SMCs are structurally independent. In Fig. 10, these SMCs are also considered independent from a probabilistic perspective in a Bayesian framework. Hence, one can say that $P(H_3) = P(SMC001-POS02 \cap SMC002-POS02) = 0.8 \times 0.1 = 0.08$. The other probabilities are similarly obtained. Note also that the construction order for the hypothesis tree (i.e., $SMC001$ first or $SMC002$ first) doesn’t matter; the two trees tell the same story.

In the example being used here, although *SMC001* and *SMC002* are structurally independent, one could still formulate, from a probabilistic perspective, sentences like:

- « If *SMC001* is *SMC001-POS01*, there is a 90% chance of having *SMC002-POS01*, and a 10% chance of having *SMC002-POS2* »
- « If *SMC001* is *SMC001-POS2*, there is 0.01 chance of having *SMC002-POS01*, and 0.99 chance of having *SMC002-POS2* »

Hence, in this second case, *SMC001* and *SMC002* are structurally independent from a “hypothesis tree” graphical representation perspective, but they are dependent from the perspective of probabilities. This is illustrated in Fig. 11. In this case, $P(H_3) = P(SMC001-POS2 \cap SMC002-POS2) = P(SMC001-POS2) P(SMC002-POS2 | SMC001-POS2) = 0.8 \times 0.99 = 0.79$. As shown on Fig. 11, one can also establish in this case that $P(SMC002-POS2) = P(H_3 \cup H_4) = P(H_3) + P(H_4) = 0.79 + 0.02 = 0.81$.

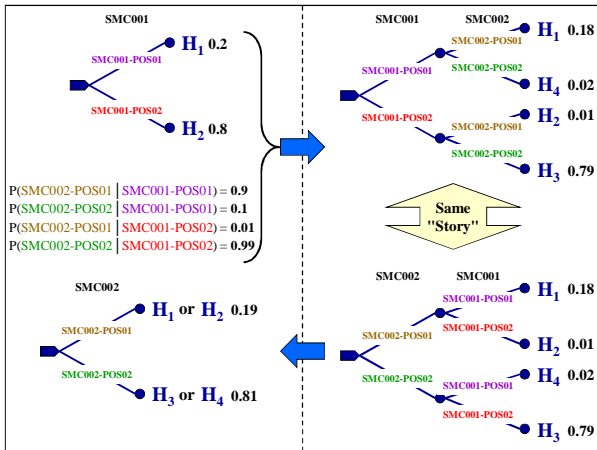


Figure 11: Computing probabilities – dependent case

9 MHA support system prototype

A proof-of-concept prototype of a multiple hypothesis situation analysis support system (MHA-SS) has been developed at DRDC Valcartier [3, 4]. The key objective for this prototype is to showcase the potential and utility of MHA. It has thus been conceived to allow users, developers, and managers to better grasp all aspects of the MHA process previously described in this paper.

9.1 Display and user interactions

Figure 12 shows the user interaction interface of the MHA-SS prototype [3]. The top part of Fig. 12 shows the *SM Bubble Display* that is used to explore the “bubbles and links” graphical representation of the situation. The middle part of Fig. 12 shows the *SM Components View* panel of the MHA-SS interface. The information content of this panel, although provided in tabular format, is equivalent to that of the “bubbles and links” and “hypothesis tree” representations.

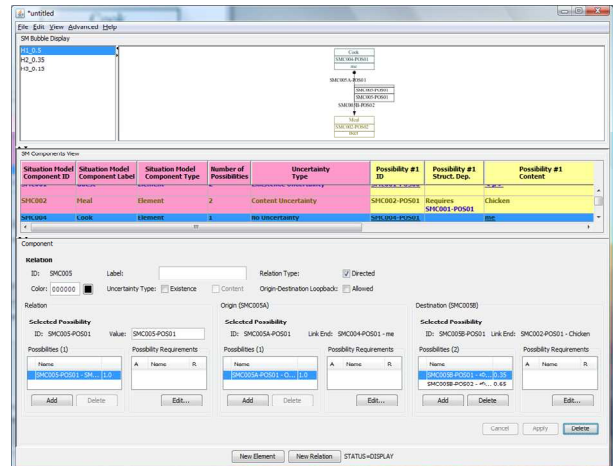


Figure 12: User interaction interface of the MHA-SS

Figure 13 shows the MHA-SS display used to explore the “hypothesis tree” graphical representation.

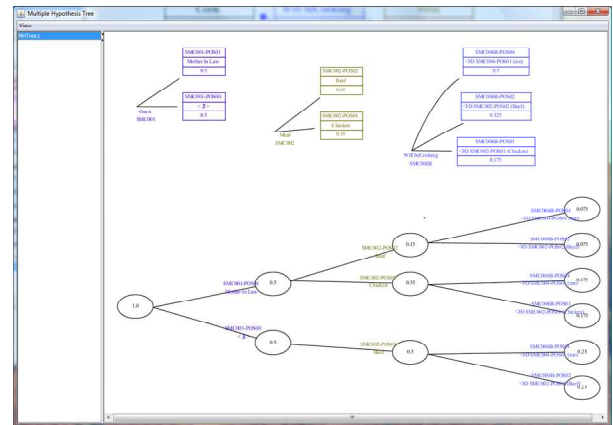


Figure 13: Hypothesis tree display of the MHA-SS

Regarding structural dependencies, if the user wants a given possibility of a component to *require/affect* the possibilities of some other component(s), he/she can open the *Requires Editor* panel for this possibility (Fig. 14). This panel shows the possibility for which the dependency requirements are to be specified. These requirements are regrouped by component. If the possibility being considered requires another possibility, one has to set the *required status* to *True* or *False* for the concerned possibility. If the possibility being considered 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. If it is in italics, then it has been implicitly defined because of a *transitive requires*. If it is bold but disabled, then it is a *system requires* (e.g., a *Relation Destination Connecting Point* that inherently requires the target of the relation, since this relation destination point cannot exist without its target). If the checkbox is unselected and disabled, then it means this is not allowed, because it would cause the possibility to be *required true* and *required false* at the same time.

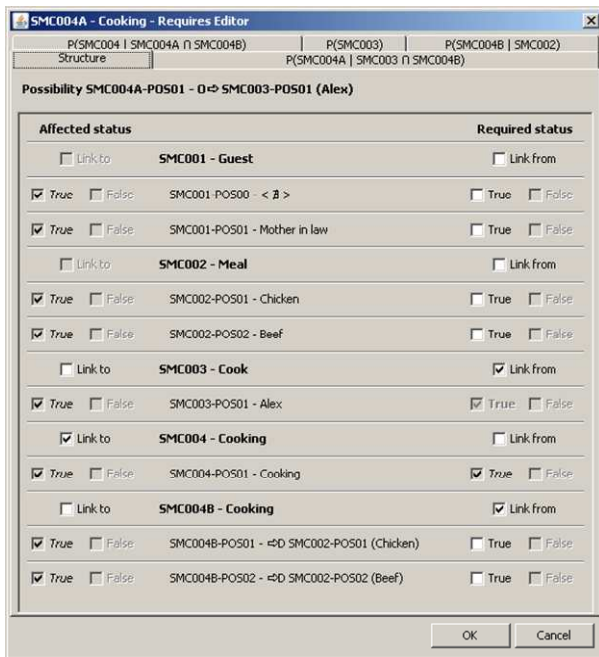


Figure 14: Possibility dependency requirements editor

9.2 Uncertainty manager

A modular *Uncertainty Manager* has been designed for the MHSA-SS, making use of generic *uncertainty containers*. So far however, only a probability-based model (Bayesian framework) has been implemented. Other approaches will eventually be considered (e.g., evidence, fuzzy-set, possibility, and rough-set theories).

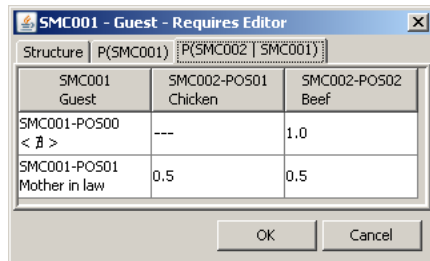


Figure 15: Probability definition tab

Structural dependencies impose probability dependencies between components and force some conditional probability values to “0” or “1”. However, the user may create additional probability dependencies between components and provide the required conditional probability values. Within the *Structure* tab of the *Requires Editor* window (Fig. 14), checkboxes allow to define links between components. These links are used to define the probabilities in the Bayesian framework. Loopbacks are not allowed, i.e., the probability dependency “path” is not allowed to become cyclic. 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 *Requires Editor* window is the *Structure* tab. However, other tabs exist to allow the user to edit the probabilities for each possibility, as defined by the *links* between the components (Figs. 14 and 15). Each row must be equal to “1”, and if the probability is “---”, then it cannot be edited. Default independent and conditional probabilities are given uniform distribution (when not already forced by the system to be “0” or “1”).

10 Conclusion

Multiple hypothesis situation analysis has been proposed as a means to provide support to the intelligence staff having to deal with uncertainty in situation analysis. A proof-of-concept prototype of a MHSA support system has been developed with the objective to showcase the potential and utility of MHSA. It has been conceived to allow users, developers, and managers to better understand each and every aspect of the MHSA process.

This paper discussed a situation modeling graphical language, the interdependency and uncertainty about the situation model components, the hypothesis tree data structure used to keep track of the uncertainty, different issues regarding the hypothesis tree (with emphasis given to the identification and management of the explicit and implicit dependencies that arise from the relationships between the situation model components), hypothesis scoring, and user interactions with the MHSA-SS.

Preliminary results have demonstrated the potential value of the MHSA approach. As these results look promising, a survey of existing link/situation analysis tools used by the intelligence community has been performed, and an evaluation has been made as to how the MHSA-SS prototype could be integrated into such tools. This could be the next R&D step for MHSA. Other potential future work being considered is related to the integration of automated reasoning within the MHSA framework.

References

- [1] Roy, J., *Towards Multiple Hypothesis Situation Analysis*, Proceedings of the 10th International Conference on Information Fusion (Fusion 2007), Quebec, Canada, July 9-12, 2007.
- [2] Bossé, É., Roy, J. and Wark, S., *Concepts, Models and Tools for Information Fusion*, Artech House, Norwood, MA, 2007.
- [3] Duclos-Hindié, N., *Multiple Hypothesis Situation Analysis – User Guide*, Thales Systems Canada, DRDC Scientific Authority: Alexandre Bergeron Guyard, DRDC Valcartier Document Number CR 2009-328, 31 August 2009.
- [4] Duclos-Hindié, N., *Multiple Hypothesis Situation Analysis – Requires/Affects Manager*, Thales Systems Canada, Contract No: W7701082823/001/QCL, DRDC Scientific Authority: Alexandre Bergeron Guyard, Document Control No: 1868C.007ReqAffManager, Rev.01, 15 April 2009.