Computational Creativity for Counterdeception in Information Fusion

Magnus Jändel
Swedish Defence Research Agency
Stockholm, Sweden
magnus.jaendel@foi.se

***Abstract*—Setting out from a discussion on why information fusion systems are vulnerable to deception, we note that exposing deception requires creativity. The burgeoning field of computational creativity is briefly reviewed and we sketch a high-level architecture for enhancing information fusion systems with computational creativity agents for the purpose of increasing robustness against deception. A more in depth exposition of how computational creativity integrates with a situation assessment sub-system based on statistical relational learning is provided and we note similarities between the different modes of this configuration and common human creativity techniques. One of these techniques is studied in a detailed example and used for revealing a suitable technical format for the creative input. Furthermore, we study how creative inputs are generated based on conceptual blending analysis applied to a library of deception stratagems.**

Keywords-deception; information fusion; computational creativity; statistical relational learning; conceptual blending theory

I. INTRODUCTION

Modern information fusion systems resemble the human brain in that they both are hierarchical pattern recognition systems. Sensors and other data collectors provide raw data feeding several consecutive fusion layers. Pattern recognizers in each successive layer receive symbols from lower layers and produce fused symbols that are passed on to higher layers. Crucially, there are also feedbacks from higher levels to lower levels that are used for telling the lower levels what to expect and what to look out for. Sub-systems learn from data and from operator feedback. The perception system of the human brain operates according to the same principles as advanced information fusion systems [1] partly because nature has inspired technology and partly because this is a reasonable way to organize a robust pattern recognition system. Airborne chemical compounds are for example captured by sensors in the nose; sensor data are preprocessed in the olfactory bulb and then passed on to specialized pattern recognizers in the olfactory cortex. Higher-level cortical systems fuse olfactory symbols with corresponding information stemming from visual, auditory, touch and proprioceptive senses. A characteristic feature of the brain is that feedback connections from higher processing levels to lower levels are very important and sometimes carries more than ten times as much data as in the forward direction [2].

A common feature of technical and biological hierarchical pattern recognition systems is that they can be fooled by mirages and illusions. Wind turbines can be perceived as approaching aircraft in airport radar systems [3] and the brain is also easily fooled by its emotion-controlled pattern recognition [4] and may for example at the first glance mistake a bicycle inner tube for a snake. The power and sophistication of the pattern recognition systems make them error-prone in unusual situations and hence vulnerable to deception.

Deception is the art of confusing the opponent´s pattern recognition systems so that false perceptions promote actions that serve the interests of the deception maker. History shows that deception wins battles and campaigns. The Mongols deftly applied deception for defeating European forces in a series of battles 1222-1242 AD [5]. Intelligence operations revealed that European soldiers viewed retreat from the battleground as a surefire signature of weakness and disarray. The Mongol general Batu therefore used faked flight as a standard stratagem. European forces in prepared positions invariably took feigned retreat as a signal for general pursuit thus transforming themselves to perfect piecemeal targets for the agile Mongolian horse archers.

So how do we handle deception in information fusion systems? For a modern and comprehensive review, we refer to Bennet and Waltz [6] and will otherwise just mention a few representative examples from the literature on deception and information fusion. Reference [7] describes a fusion system that discovers tactical deception by identifying inconsistencies including objects apparently moving with impossible speed. Recognized anomalies are used for weeding out deceptive information either automatically or by escalating to human attention. Reference [8] applies machine learning to exposing deception in text data. Trained on case histories from crimes on military bases the system achieved better than 74% classification accuracy. Inconsistency detection as in [7] and learning from incidents as in [8] should be combined in advanced counterdeception systems.

Uncovering deception is an act of creativity. Deception exploits the victim´s habitual pattern recognition and reasoning processes. Bennet and Waltz encourage analysts to "… *break loose of potentially blinding mindsets in order to see the situation in a completely different light*" [6]. Snapping out of time-honored cognitive patterns is the hallmark of creativity. Aiming for at least partial automation of creativity, this paper

This research is supported by the R&D programme of the Swedish Armed Forces.

explores how computational creativity techniques can be employed in information fusion systems.

Section II succinctly reviews computational creativity focusing on the relevance for counterdeception and expanding on the technique of conceptual blending. Section III introduces a high-level architecture for extending a generic information fusion system with computational creativity agents. The situation assessment level is considered in greater detail in section IV. Section V describes the nuts and bolts of one of the creativity methods from section IV. Discussion and conclusions are offered in section VI.

II. COMPUTATIONAL CREATIVITY

A. Introduction to Computational Creativity

Computational Creativity is a branch of Artificial Intelligence that is concerned with two main research questions: 1) how to build creative machines and 2) how to mechanistically explain human and animal creativity. Both of these issues have a long history and scattered results are found in the annals of computer science, cognitive science, philosophy, psychology and neuroscience [9]. Recently computational creativity has found a home in the conference series International Conference of Computational Creativity (ICCC). Browsing the proceedings of ICCC [10], [11], [12] is a good starting point for learning about modern computational creativity. Computational creativity has a foundational branch and an applied branch where the former tries to provide definitions and measures of creativity while the latter strives to demonstrate computational creativity in various fields including product innovation, marketing, story generation, poetry, music and art. There are also a few applications to decision support [13], [14]. A wide range of AI methods are used in computational creativity including logic, planning algorithms, evolutionary algorithms, and multi-agent programming. The next sub-section reviews a prominent computational creativity methodology that is relevant for analyzing deception and is used for designing a counterdeception algorithm in section V.

B. Conceptual Blending Theory and Deception

A deception scheme is a blend between the *revealed situation* and the *concealed situation* [15] where the deceiver intends to mislead the victim into perceiving the revealed situation while surreptitiously performing the concealed operation. Conceptual Blending Theory (CBT) and in particular Double-Scope Blending (DSB) are useful both for inventing deception strategies and for uncovering deception. Fauconnier and Turner provide a comprehensive account of both CBT and DSB [16]. Briefly, DSB means mixing two input mental spaces to produce a third blended mental space (see Fig. 1). A generic mental space guides the mapping between the input spaces. The CBT of Fauconnier and Turner is a psychological explanatory theory in which the concept of mental space is rather loosely defined as the system of concepts and relations that are related to some context or topic. Computer scientists can think of mental spaces as taxonomies or ontologies and implement the blending process as taxonomy or ontology operations.

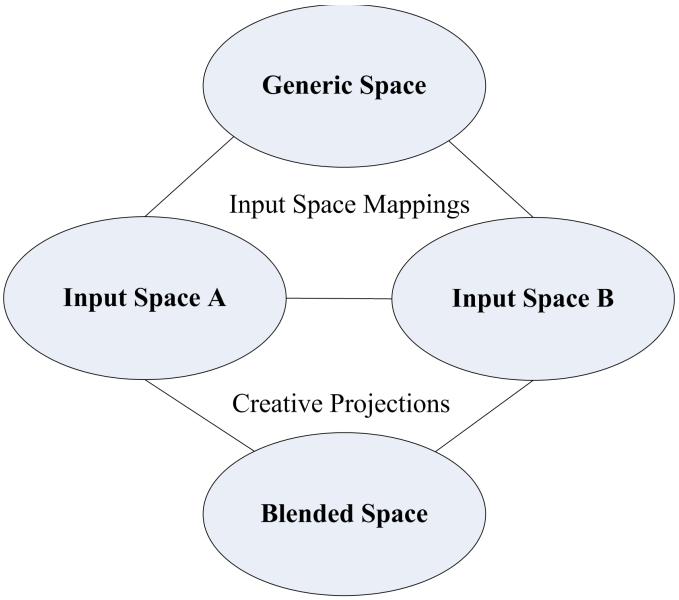

Fig. 1. Double-scope conceptual blending. The mapping of Input Space A and B into the Blended Space is guided by the common ground of the Generic Space.

As a simple example of DSB we analyze an archetypical deception operation – the Trojan horse. The generic space is **Transportation**; input space A is **Gifts**; input space B is **Invasion**; the blended space is the **Trojan horse operation**. The concept *Vehicle* from the generic space is mapped both to the concept of a *Gift* in in Input space A and to the concept of *Means of transportation for the invading force* in Input space B. The blended space includes the Trojan horse prototype - a concept that is the confluence of *Gift* and *Means of transportation for the invading force.* The CBT process includes a phase of elaboration of the initial mapping. Elaboration may include pattern completion, fleshing out details and further mappings. In the Trojan horse case, elaboration contributes that the gift is shaped as a gigantic wooden horse, that it will be filled with an elite cadre of warriors, that the invading fleet feigns sailing home etc. There are no rights or wrongs in the blending process – it is an act of creativity.

Conceptual blending is used for both construction and analysis. The work of Fauconnier and Turner and the lion's share of the literature on CBT are analytical focusing on explaining phenomena in psychology, linguistics, society and the arts. Conceptual blending has also been used constructively as for example in the work of Tan and Kwok [13] in which DSB is applied to scenario planning for maritime security in Singapore. Tan and Kwok use **Shipping** as the generic mental space, **Peaceful Shipping** and **Maritime Terrorism** as mental spaces A and B respectively and construct blends that serve to make security forces aware of possible deception strategies. Tan and Darken also applied CBT to prediction in high-level fusion [17].

For constructing a deception strategy with CBT, select the concealed operation that you wish to perform as one of the input spaces. The other input space will be the revealed situation that you want the victim to perceive. The generic space is a top ontology that connects the input spaces. The

blended space is the deception stratagem. In counterdeception let one of the input spaces be the situation as presently perceived. The other input space is a deception stratagem. The blended space is the concealed situation. Elaborate the blend to match current observations and use it to suggest telltale signals.

III. High-Level Architecture

Fig. 2 outlines a top-level architecture for enhancing an information fusion system with auxiliary computational creativity functions. We assume that the base-line *information fusion system* (IFS) is a legacy product that conceptually is structured according the JDL model [18] and operates as outlined in the introduction. *Computational Creativity Agents* (CCA) receive selected data from the IFS and input *creative hypotheses* (CH) to the different levels of the JDL hierarchy. The creative input may cause major shifts in situation and threat assessment. The Assessment Arbiter (AA) decides on which assessment to keep. The AA can be a human decision-maker or a very advanced software agent. Creating new deception hypotheses falls in the cognitive domain of the conceptual hierarchy of data fusion functions that was introduced by Waltz [19] and cognitive-level intelligence is required for resolving competing situational assessments.

The IFS runs the standard information fusion algorithm for most of the time. While doing that, it is, as discussed in section I, vulnerable to deception. The enhanced system switches occasionally to a mode where it accepts a CH and explores its consequences as outlined in Box 1. We assume that the IFS works with hypotheses expressed in a logic-based language and that the CH is expressed in the same language. This is not a strictly necessary assumption but it facilitates the further discussion.

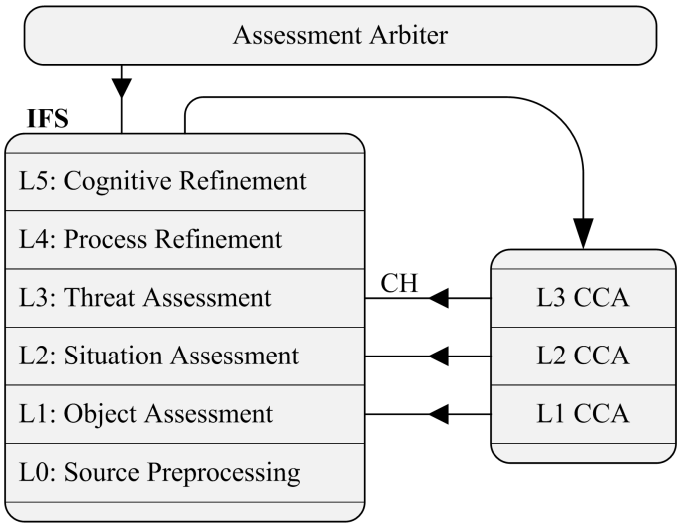

Fig. 2. Creativity-enhanced information fusion system (IFS). Computational Creativity Agents (CCA) provide creative hypotheses (CH) to the IFS guided by selected IFS data.

1) Save the state of the IFS
2) Insert the CH at the appropiate JDL level which will be called the *insertion level*.
3) Resolve any inconsistencies by disabling legacy hypotheses and rules that contradict the CH.
4) Run the IFS for a predetermined time. The new state of the insertion level influences both higher and lower levels leading to a cascade of changes that ripples through the IFS until a new stable situation and threat assessment forms or the allotted time runs out.
5) If the CH has catalyzed a novel situation and threat assessment, the AA decides which of the assessments to keep. The IFS will either restore the state that was saved in step 1 or continue operating in the new state.

Box 1. Process for integrating and exploiting a CH.

It is conceivable to apply computational creativity at all JDL levels but we will for brevity only discuss the effects at levels 1-3.

Level 1: At the object assessment level, the CCA could suggest alternative identifications of objects that may serve to uncover camouflage. Using IR sensor data, the regular IFS may for example have identified a vehicle as a noncombatant truck but a CH forces the system to tentatively label the object as an enemy armored vehicle. This will cause the higher layers to reevaluate the situation and threat model and could also trigger the process refinement layer to direct high-resolution sensors to the object.

Level 2: At the situation assessment level, the CCA could suggest alternative complete or partial situation assessments. An early-warning IFS may for example under the influence of intelligence reports interpret target patterns as a previously announced scientific multi-stage rocket launch. The CH could induce the system to consider the alternative explanation of a hostile missile launch and the subsequent splitting of the missile into several parts indicating a MIRV warhead. The CCA might be inspired by reports of an event [20] where a scientific rocket briefly was interpreted as a possible missile launch and infer a deception stratagem by reversing the actual circumstances.

Level 3: At the impact assessment level, the CCA could suggest a different interpretation of the threat or novel threat modes. A CCA could for example note that the IFS focuses on kinetic threats related to an enemy air strike and generate a CH causing the IFS to consider the possibility of a supporting cyber-attack on the communication system that connects early-warning radars to C&C systems. Historical records [21] might have inspired the cyber-attack option.

IV. Creative Situation Assessment

This section takes a closer look at creativity enhanced situation assessment. We consider a system according to section III where JDL level 2 is implemented using Bayesian Logic Programs (BLP) [22] which is one of the main

techniques in Statistical Relational Learning (SRL) [23]. Going to this level of detail will clarify how CHs can be integrated in the IFS and reveal a taxonomy of creativity techniques.

A. Level 2 Architecture and Normal Operation

The architecture of the subsystem is shown in Fig. 3. We will first explain the normal operation of the subsystem. The input from higher levels of the system is a Query expressed as a first-order logic predicate. The Query codifies the primary objective of situation assessment. In an early-warning air defense system, the Query might be the predicate **AirAttack**. The top-level output of level 2 is the estimated probability for the truth of the Query. In practice, there are much qualifying and supporting information in both the input and the output but we will keep things simple here. The inputs from JDL level 1 are a set of observed objects, object attributes and related measures of uncertainty. This information is stored as logical predicates with associated probabilities in the knowledge base KB_G. Level 1 updates continuously the content of KB_G. The predicates in KB_G are known as *ground terms* in the parlance of BLP.

The level 2 subsystem includes two other knowledge bases KB_L and KB_{BC} where the former holds *logical atoms* and the latter *Bayesian clauses*. Logical atoms are first-order logic sentences that are deemed to be definite and unalterable truths about the domain as defined by domain experts. Bayesian clauses express generic but uncertain and possibly changeable relations in the domain. The relational part of a Bayesian

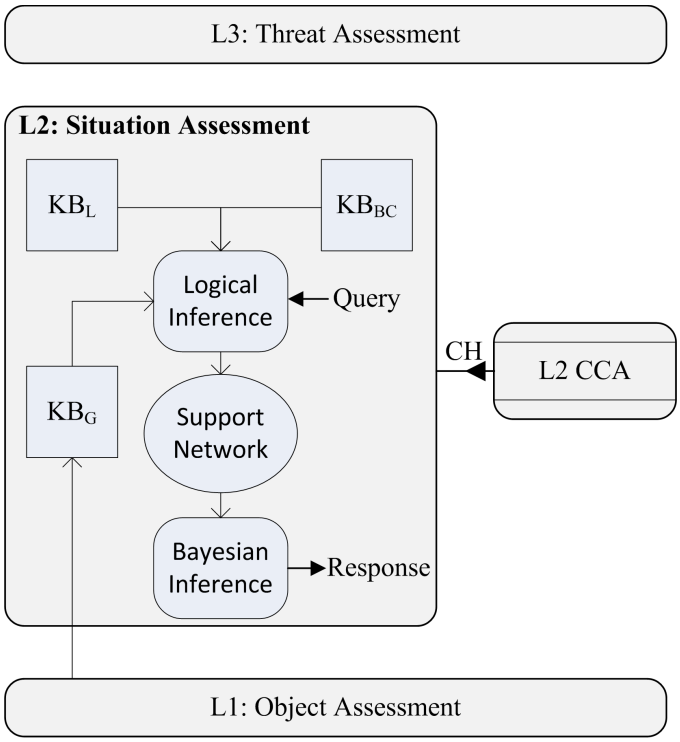

Fig. 3. Architecture for creativity-enhanced situation assessment.

clause is a universally quantified Horn clause. A Horn clause is an implication with precisely one head predicate that is implied by a conjunction of body predicates. Uncertainty is represented by a conditional probability table (CPT) associated with each Bayesian clause. There are several options for how to build the KB_{BC}. Domain experts can specify both the relational part and the CPTs of the Bayesian clauses. It is sometimes feasible to learn the CPTs from data as long as experts provide the Horn clauses and a sufficient amount of training data is available. Learning both the logical structure and the CPTs from data is a vibrant research topic but might presently not be practical in complex real-life systems (For an introduction to BLP learning see [24]).

Each situation assessment iteration starts by the Logical Inference Engine (LIE) constructing a proof of the Query using the merged logical content of the knowledge bases KB_G, KB_L and KB_{BC}. Assuming that precisely one such proof can be found, the output of the LIE is a tree representation of the proof with the Query predicate as the top node, ground terms as leaf nodes and edges corresponding to relations found in the KB_L and KB_{BC}. This *logical support network* encodes how the generic laws in KB_L and KB_{BC}, applied to the specific facts about the current situation in KB_G, proves the Query. The Bayesian Inference Engine (BIE) reinterprets the logical support network as a *Bayesian support network* with predicate symbols taken to mean stochastic variables. Bayesian clauses are instantiated as sub-trees with the head of the Horn clause mapped to the parent node and the body variables as child nodes. The CPT of the sub-tree is inherited from the source Bayesian clause. A Bayesian clause can hence be understood as a repeated pattern in the vast family of Bayesian support networks that describes all possible situations. Sub-trees stemming from logical atoms have edges with no uncertainty. Once the Bayesian support network has been constructed, the BIE computes the truth-value of the Query by Bayesian inference. In the simple IFS that we are discussing, this is the situation assessment.

For the sake of brevity, we have ignored important particulars of BLP including how to use combination rules [22] for handling cases where the LIE produces multiple proofs and how to ensure that logical support networks can be mapped to legitimate Bayesian networks. For a comprehensive account of BLP we refer to [22] and [24].

B. Applying Creative Hypotheses to Situation Assessment

This sub-section describes the impact of CHs on the situation analysis system described in the previous subsection. A CH can be *productive* or *insipid*. A productive CH stimulates the IFS to produce a different support network leading to a new situation assessment. An insipid CH causes the IFS to either generate the same response as before or alternatively to fail to find a valid support network. Productive CHs provoke results that humans deem to be either *creative* or *trivial*. The goal of the CCA is to generate productive CHs and creative results.

Table I provides an overview of the different modes of creative enhancement in the architecture of Fig. 3. Technical effects are mapped to analogous human creativity techniques.

TABLE I. TECHNICAL EFFECTS OF CREATIVE INPUT TO THE SYSTEM OF FIG. 3 VERSUS HUMAN-LEVEL CREATIVITY TECHNIQUES.

Technical effects	Creativity techniques
Remove logical atoms or Bayesian clauses. Remove predicates from the body of logical atoms or Bayesian clauses.	Escaping
Add logical atoms or Bayesian clauses. Add predicates to the body of logical atoms or Bayesian clauses.	Blocking
Change the CPT of a Bayesian clause.	Counter-empirical probabilistic reasoning
Add or remove objects from KB_G.	Counterfactual reasoning
Change uncertainty parameters of objects in the KB_G.	Counterfactual probabilistic reasoning
Change the Query.	Redefine the problem

Table I shows that creative input effects on the situation assessment system often are tantamount to human-level creativity methods (se e.g. [25] for an introduction). Well-known methods from the creativity literature are:

- The *Escaping* method in which we deliberately overlook some requirement, condition, law or constraint and harvest the resulting ideas.

- The *Blocking* method where the problem is made harder by introducing further requirements and constraints typically with the effect of inhibiting the normal solution hence forcing a search for new ideas.

- *Counterfactual reasoning* where deliberately ignoring facts about the situation stirs new ideas.

- *Redefine the problem* in which we get a new perspective on the situation by reformulating the problem description sometimes with the intention of returning to the initial problem with fresh ideas.

In Table I we also note two creativity techniques that we have not found in the literature:

- *Counterfactual probabilistic reasoning* which is similar to counterfactual reasoning with the twist that we imagine that in fact probable things and events are improbable or vice versa.

- *Counter-empirical probabilistic reasoning* where we disregard empirically known statistical relations by assuming that an in fact common casual effect is unusual or vice versa.

V. ESCAPING DECEPTION

This section explores how the Escaping technique can be used for uncovering deception in the system that is conceptually outlined in section III and specialized to situation assessment in section IV. Consider a situation in which the *blue force* IFS monitors *red force* air operations. We assume that level 1 of the IFS provides input in the form of the following references to observations (BLP ground terms). Radar surveillance provides a list of observations,

$$\mathbf{O} = \{O_1, O_2, ...O_{n_O}\}, \quad (1)$$

where each item represents a target and bold symbols indicate lists. A set of intelligence reports is represented by,

$$\mathbf{R} = \{R_1, R_2, ...R_{n_R}\}, \quad (2)$$

in which each element R_i points to a specific report. A set of meteorological reports,

$$\mathbf{M} = \{M_1, M_2, ...M_{n_M}\}, \quad (3)$$

is also provided. Real-life IFSs would also process many other types of observations including ground terms referring to electronic warfare and communications. For this example however, the ground terms listed here will suffice.

Each of the ground terms in (1), (2) and (3) denotes an information item with many attributes. Attributes are in BLP represented by predicates. Each predicate has a domain of values and associated probabilistic parameters. Level 1 of the IFS may for example estimate that the air craft type related to radar object O_{23} is Type(O_{23})=JAS_39 with probability 86 % where Type is a predicate and JAS_39 is a value in the domain of the predicate. For brevity, we will not further define the panoply of predicates that the system uses for representing and reasoning with ground term attributes.

The general situation is described by high-level predicates, here simplified as follows. The overall target situation is described by the predicate $T(\mathbf{O}) \in \mathfrak{D}_T$ which is a function of the radar objects and has values in the domain $\mathfrak{D}_T$. The elements of $\mathfrak{D}_T$ are high-level assessments tantamount to plain language statements such as for example: *A squadron of red force fighters is taking off*. Weather conditions are similarly described by the predicate $W(\mathbf{M}) \in \mathfrak{D}_W$ where the domain $\mathfrak{D}_W$ for example includes values corresponding to plain language: *Clear and sunny weather*. The IFS refines intelligence reports to high-level conclusions about red force air-related activities. Here, we simplify this to conclusions on red force exercise activities and on the prospect for air attacks as represented by predicates $I_E(\mathbf{R}) \in \mathfrak{D}_E$ and $I_A(\mathbf{R}) \in \mathfrak{D}_A$ respectively. Note that I_E and I_A are based only on intelligence reports and ignores radar and meteorological observations.

The final level 2 conclusions on red force air exercises and attacks are based on joint analysis of all the high level predicates as expressed by the following Bayesian clauses,

$$E :- I_E, T, W, \tag{4}$$

$$A :- E, I_A, T, W, \tag{5}$$

in which we use Prolog-type notation [26] so that (4) means that E is logically implied by the conjunction of I_E, T and W. The top-level binary predicates E and A indicate that current red force air operations are interpreted as exercise and attack respectively. The CPTs that are associated with the top-level Bayesian clauses enables the system to compute exercise and attack probabilities.

The ground terms, predicates and Bayesian clauses defined so far form the foundation for computing the Bayesian support network for situation analysis as described in section IV and illustrated in Fig. 4. Note that ground terms and predicates are interpreted as stochastic variables in the Bayesian support network. We use the same symbol in both interpretations so that for example T(**O**) stands for a predicate in the logical context and a stochastic variable in the Bayesian context. There are three domain-specific Bayesian subnets for processing object observations (**O**), meteorological reports (**M**) and intelligence reports (**R**) respectively. Each such subnet takes the associated set of ground term variables as input and outputs high-level variables.

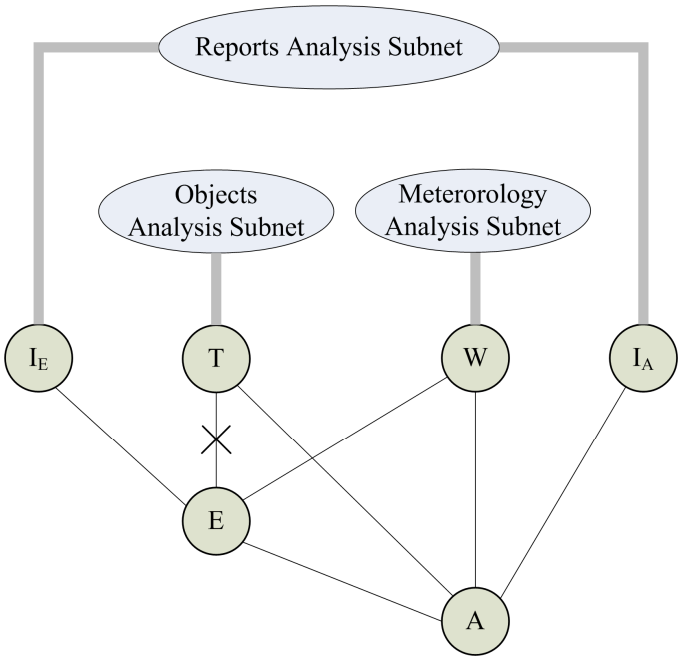

Fig 4. Bayesian support network for situation analysis. The object analysis subnet processes object observations **O** and outputs the stochastic variable T. Similarly, the meteorology analysis subnet creates W. The intelligence analysis subnet produces the variables I_E and I_A. Equations (4) and (5) give rise to the lower part of the support network that produces the exercise indicator E and attack indicator A. The crossed-over edge between T and E indicates the effect of a CH.

The subnets are based on a rich set of domain specific Bayesian clauses and include appropriate generic knowledge from KB_L. The subnets are therefore quite complex and incorporate many specialized intermediate stochastic variables that are not shown in Fig. 4. Note that the subnets are dynamic since they are constructed by logical proof of the output predicates based on a dynamically shifting set of ground term variables. The output of the subnets are the high-level variables T(**O**), W(**M**), I_E(**R**) and I_A(**R**) where the two latter both are produced by the intelligence analysis subnet. Furthermore, Fig. 4 shows how the top-level Bayesian clauses (4) and (5) are cast into the high-level part of the Bayesian support network. The probabilities for that the red force is engaged in air exercise or attack are produced by Bayesian inference in the support network of Fig. 4.

Assume now that the system finds that the red force with high probability is exercising and that the risk for attack is negligible. The CCA applies the Escaping technique to produce a CH that prompts the IFS to reconsider. We will first discuss the format and effect of the CH and then outline how it can be created. The CH is a command to the IFS codified as,

$$At \;\; \&(E :- I_E, T, W) \; replace \; with \; E :- I_E, W. \tag{6}$$

The IFS interprets this as an instruction to go to the address of the expression $E :- I_E, T, W$ in KB_{BC} and replace the logical part of the Bayesian clause found there with the expression $E :- I_E, W$. The CPT of the new clause is computed by summing out the influence of the removed predicate T. The effect of the CH is to make the IFS ignore radar observations of red force air operations when estimating the likelihood that an exercise is going on. This is marked in Fig. 4 by crossing over the link between the T and the E nodes. The CH thus makes the IFS forget the empirically knowledge that the current red force operational pattern is related to exercises. This is tantamount to applying the creativity technique of Escaping that was described in sub-section IV B.

The new structure of the support network triggers a re-calculation of the output in which only intelligence reports and weather conditions are taken into account for computing the probability of an air exercise. Since A depends on E this will also change the estimated probability of air attack, which could be instrumental for exposing deception. To see why, we will now discuss how the CCA might come up with the CH. The interfaces of a CCA that could produce the CH of (6) are outlined in Fig. 5. The CCA examines the current support network from the IFS and notes that the situation is judged to be peaceful with an ongoing exercise and a very low probability for imminent attack. Furthermore, the CCA analyzes intelligent reports that indicate rising political tensions. The combination of an apparently peaceful operational picture and underlying strategic conflict is a signature of possible deception.

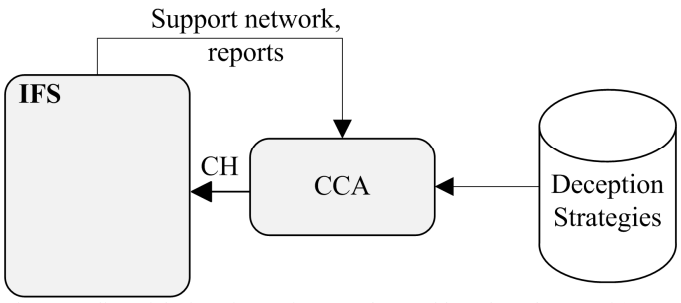

Fig. 5. Outline of the interfaces of a CCA that could produce the CH of (6) . It receives the support network and intelligence reports from the IFS. An archive of deception strategies is available.

The CCA proceeds by browsing an archive of deception strategies that are expressed in first-order logic and thus forms a separate knowledge base. Note that this archive is similar to the *Basic D&D templates* that are described in a more generic analytical context by Bennet and Waltz [6]. By pattern matching or even crude random search, the CCA homes in on deception stratagem **43**, here expressed in English as:

A. Associate an attack scheme AP with training behavior by long-term habituation where AP is used in a series of exercises.
B. Launch a surprise attack using attack scheme AP.

This stratagem is well known since antiquity and has since then been constantly adapted and applied to new domains [27]. The CCA compares deception stratagem **43** to the current situation by applying a formalization of the CBT-analysis that was described in sub-section II B (see Fig. 6).

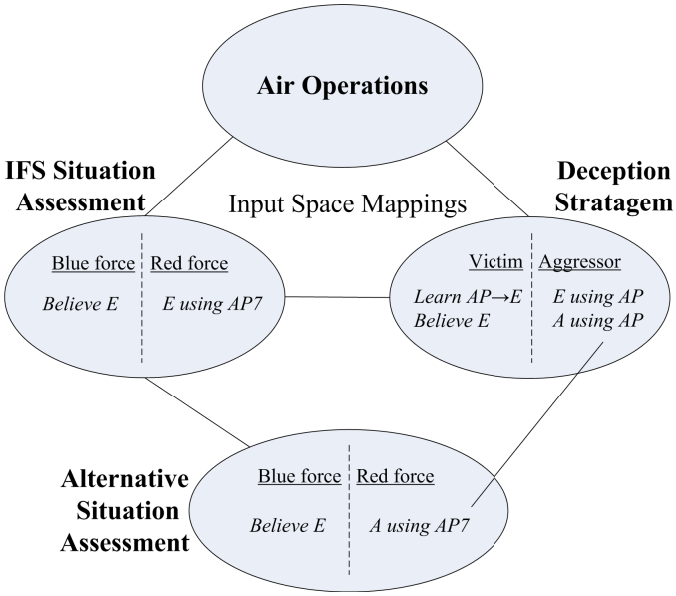

Fig. 6. CBT-analysis performed by the CCA. The top ontology **Air Operations** facilitates creative mapping between the two input spaces **IFS Situation Assessment** and **Deception Stratagem**. Symbols E and A mean Exercise and Attack respectively. The Blue force is the Victim and the Red force is the Aggressor. The current situation is mapped to the final phase of the deception stratagem. The presently used red force attack pattern AP7 is mapped to the generic attack pattern AP of the deception stratagem. The blended space **Alternative Situation Assessment** indicates that the red force performs a surprise attack employing AP7.

Based on the CBT analysis, the CCA searches the present support network for an edge that serves as a channel for communicating the revealed situation according to the deception stratagem. In the present example, this would be an edge that carries the association between attack pattern AP7 and red force exercises and fulfills the following criteria:

a) The edge is critical for inferring the current conclusion that no attack is expected (to achieve novelty in the new assessment)
b) The CPT of the edge is conditioned on historical red force tactical behavior (to match step A of the deception stratagem).
c) Ground terms connected to the edge are related to the current red force tactical behavior (to match step B of the deception stratagem).

The edge between nodes *T* and *E* satisfies all these conditions (see Fig. 4). Node *T* influences node *E* that in turn affects node *A*. The statistical parameters of the edge are based on learning associations between tactical behavior and exercises. The *T* variable is, via the object analysis subnet, connected to ground terms related to radar observations of current air operations. Based on this analysis, the CCA sends a CH according to (6). The CCA may also compose an explanatory message intended for the operator reading: "*Assuming that the opponent executes deception strategy **43** it is suggested that data on current air operations are ignored when estimating the likelihood of a red force exercise*". Note that the selected edge might not be unique in fulfilling criteria *a-c*. The CCA could select a suitable edge randomly or be biased to favor edges between high-level variables since it is easier to concoct human-level explanations for CHs that impact on higher levels of the support network.

If the opponent actually is launching a surprise attack under cover of deception, the CH could influence the IFS to give higher weight to intelligence reports that may carry indications of the impending aggression. This will cause the system to revise the probabilities of the top-level variables, which in turn may increase decision-maker awareness of the possibility of deception.

VI. Discussion and Conclusions

Why are the CCAs separate systems and not integrated in the IFS? Integration is a viable option but keeping the CCAs at arms-length from the IFS has distinct advantages. We can make the analogy with human enterprises where it is prudent to have a stable and perhaps a bit unadventurous core management. To drive change, senior management may from time to time recruit creative consultants or encourage internal hotheads to pioneer new ideas. The final decision on whether or not to make a major change should always rest on senior management advised by both conservatives and radicals. Organizatorial merging of firebrands with the run-of-the-mill

staff is not optimal for attaining the right mix of stability and creative drive. Similarly, it is judicious to keep the legacy IFS stable and secure and only allow thoroughly verified and validated modifications. The CCAs are optional extensions that may be supplied by different providers and does not require the same level of security as the IFS. They can be substituted depending on operational requirements and the contingent need for counterdeception support.

Many alternative system configurations are possible beside the time-sharing arrangement shown in Fig. 2. Several IFSs could be used, one of which is in the standard mode while the other are driven by different CCAs. Yet another possibility is that JDL levels 2 and 3 of a single IFS are duplicated so that the high-level analysis follows several different tracks, some of which are guided by dedicated CCAs. Joint optimization of sensor control and data collection is required in this configuration. This last option would be an automated version of the Alternative Competing Hypothesis Process as described by Bennet and Waltz [6].

In conclusion, we have argued that advanced information fusion systems are vulnerable to deception and that counter-deception requires creativity. The emerging science of computational creativity could provide methods for fortifying information fusion systems against deception and section III consequently suggests an architecture for incorporating computational creativity agents in information fusion systems. By analyzing a JDL level 2 system based on statistical relational learning, we note that technical modes of integrating computational creativity correspond to human-level creativity techniques. Finally, we find that an archive of generic deception stratagems expressed in first-order logic enables counterdeception computational creativity based on conceptual blending.

Employing tools of computational creativity in information fusion systems is a promising route to complementing and supporting the chiefly human- and organizational-level counterdeception measures that are described in [6]. These new tools have the potential for going beyond anomaly detection and removal as in [7] and classification-based approaches as in [8] and ultimately at least partially automate operational and strategic counterdeception.

VII. REFERENCES

[1] G. M. Shepherd, Eds., The Synaptic Organization of the Brain, Oxford: Oxford University Press, 1990.

[2] S. M. Sherman and R. W. Guillery, Exploring the thalamus and its role in cortical function}, MIT Press, 2006.

[3] D. Jenn and C. Ton, ”Wind Turbine Radar Cross Section,” *International Journal of Antennas and Propagation,* vol. 2012, p. Article ID 252689, 2012.

[4] D. M. Eagleman, ”Visual Illusions and Neurobiology,” *Nature Reviews Neuroscience,* vol. 2, pp. 920-926, 2001.

[5] D. Sinor, ”The Mongols in the West,” *Journal of Asian History,* vol. 33, pp. 1-44, 1999.

[6] E. Bennet and E. Waltz, Counterdeception: Principles and Applications for National Security, Artech House, 2007.

[7] J. W. Choi, J. W. Joo and D. L. Cho, ”Situation/Threat Assessment Fusion System (STAFS),” in *Fifth International Conference on Information Fusion*, 2002.

[8] C. M. Fuller, D. P. Biros and D. Delen, ”An investigation of data and text mining methods for real world deception detection,” *Expert Systems with Applications,* vol. 38, nr 7, pp. 8392-8398, 2011.

[9] J. McCormack and M. d'Inverno, Eds., Computers and Creativity, Springer, 2012.

[10] ”First International Conference on Computational Creativity,” 2010.

[11] ”Second International Conference on Computational Creativity,” 2011.

[12] ”Third International Conference on Computational Creativity,” 2012.

[13] K.-M. T. Tan and K. Kwok, ”Scenario Generation Using Double-Scope Blending,” in *AAAI Fall Symposium*, 2009.

[14] M. Jändel, ”Computational Creativity in Naturalistic Decision-Making,” to appear in proceedings of the 4th International Conference on Computational Creativity,” 2013.

[15] M. Anderson, ”Cultural Concatenation of Deceit and Secrecy,” in *Perspectives on Human and Nonhuman Deceit*, R. Mitchell and N. Thompson, Eds., State University of New York Press, 1986.

[16] G. Fauconnier and M. Turner, The Way We Think: Conceptual Blending and the Mind's Hidden Complexities, Basic Books, 2002.

[17] T. Tan and J. Darken, ”Faster conceptual blending predictors on relational time series,” in *15th International Conference on Information Fusion*, 2012.

[18] A. Steinberg, C. Bowman and W. E.F., ”Revision to the JDL Data Fusion Model,” in *Joint NATO/IRIS conference*, 1998.

[19] E. Waltz, ”Data fusion in offensive and defensive information operations,” in *National Symposium on Sensor and Data Fusion*, 2000.

[20] B. Blair, H. Feiveson and F. von Hippel, ”Taking Nuclear Weapons off Hair-Trigger Alert,” *Scientific American,* vol. November, 1997.

[21] S. Applegate, ”The principle of maneuver in cyber operations,” in *4th International Conference on Cyber Conflict*, 2012.

[22] K. D. R. L. Kersting, ”Bayesian Logic Programing: Theory and Tool,” in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds., The MIT Press, 2007.

[23] L. Getoor and B. Taskar, Eds., Introduction to Statistical Relational Learning, The MIT Press, 2007.

[24] K. Kersting and L. De Raedt, ”Basic Principles of Learning Bayesian Logic Programs,” 2002.

[25] E. de Bono, Six Thinking Hats: An Essential Approach to Business Management, Little, Brown& Company, 1985.

[26] I. Bratko, Prolog programming for artificial intelligence, Addison Wesley, 2001.

[27] P-Y. Chen, S-M. Cheng, and K-C. Chen, "Smart Attacks in Smart Grid Communication Networks," IEEE Communications Magazine, August 2012.