Abstract – The paper contends that a symbolic approach to higher-level machine data fusion is required. This induces an Epistemic Challenge that is arguably best addressed through a combination of cognitive and analytic approaches. A cognitive model, a process for cognitive capture, and a computational framework for automating captured cognitive routines, are outlined.

Keywords: data fusion, JDL model, \( \lambda \)JDL model, sensor fusion, information fusion, situation assessment, impact assessment.

1 Data Fusion

1.1 The JDL Model of Data Fusion

Lambert \[1][2][3] defines data fusion as the process of utilising one or more data sources over time to assemble a representation of aspects of interest in an environment. The JDL model is currently the most widely accepted model of the data fusion process. Figure 2 illustrates the current JDL model.

Steinberg et.al. \[4\] provides definitions of the component levels.

1.2 The \( \lambda \)JDL Model of Data Fusion

Lambert \[1][2][3]\ provides a revision of Steinberg et.al.’s \[4\] revised definitions of “level 1”, “level 2” and “level 3” of the JDL model, while including “level 0” within “level 1” and absorbing “level 4” within each of the other levels. Under the author’s revised JDL account, termed \( \lambda \)JDL in \[3\] level 1 is about the identification of objects from their properties, level 2 is about the identification of relations between these objects; and level 3 is about the identification of the effects of these relationships between these objects. Lambert \[3\] states,

- **Object fusion** is the process of utilising one or more data sources over time to assemble a representation of objects of interest in an environment. An object assessment is a stored representation of objects obtained through object fusion.
- **Situation fusion** is the process of utilising one or more data sources over time to assemble a representation of relations of interest between objects of interest in an environment. A situation assessment is a stored representation of relations between objects obtained through situation fusion.
- **Impact fusion** is the process of utilising one or more data sources over time to assemble a representation of effects of situations in an environment, relative to our intentions. An impact assessment is a stored representation of effects of situations obtained through impact fusion.

Figure 2 illustrates the \( \lambda \)JDL data fusion process. Hereafter object fusion will on occasion be called sensor fusion, while situation and impact fusion will often be collectively termed information fusion.

2 Epistemic Challenge

2.1 Machine Data Fusion

Machine data fusion arises when we interpret “representation” as “machine representation” in the \( \lambda \)JDL model of section \[2\]. Machine data fusion is about having machines develop object, situation and impact assessments. This has enjoyed mixed success to date. Machine sensor fusion has matured through established techniques for detection, association and estimation. By contrast, machine based information fusion remains a fledgling enterprise.

The reason for this disparity is twofold.
1. The bulk of the data fusion community remain heirs to the traditions and techniques of sensor fusion.
2. The techniques of sensor fusion do not readily scale up to the complexities of information fusion. Lambert [1][2][3][5] observes that there is a paradigm shift as we move from sensor fusion to information fusion, a shift from an Aristotelian world of objects with measurable properties, to a Wittgensteinian world of symbolically expressed facts formed from relations between objects. The concept of a relation is less than 200 years old and arose to overcome limitations with the Aristotelian conception of objects with properties [6]. Related Aristotelian limitations are evident within the conception of the sensor fusion community [5] while the JDL definition of section 1.2 and the revised JDL account of Steinberg et.al. [4] both explicitly identify situation fusion as requiring the more sophisticated relations construct.

2.2 Semantic and Epistemic Challenges

If the techniques of traditional sensor fusion are not going to redress the complexities of information fusion, then how are we to achieve machine based information fusion? In [3] the author promotes, inter alia, the Semantic Challenge and the Epistemic Challenge for information fusion.

Semantic Challenge: What symbols should be used and how do those symbols acquire meaning?

Epistemic Challenge: What information should we represent and how should it be represented and processed within the machine?

This paper addresses aspects of the Epistemic Challenge.

The machine sensor fusion strategy has primarily promoted analytic solutions to object fusion problems. This involves developing a mathematical model of the problem environment and then formulating existence proofs, solutions or optimisations within that mathematical model. But assumptions (like Gaussian noise) and simplifications (like linearity) are often required to obtain a mathematical solution. Analytic closed form solutions are often provably unattainable (as with polynomials over degree 4). The numerical analysis calculations applied in lieu of closed form analytic solutions must often contend with intractability [7] if not undecidability [8] and so require heuristics to even secure a satisfactory solution. Because the complex problem environment rarely matches our ideal mathematical conception of it, approximations must often be made.

As the complexity of the information fusion task exceeds the complexity of the sensor fusion task, a purely analytic solution is an even more difficult proposition for the information fusion task. The central theme of this paper is to suggest cognitive solutions. Instead of developing a direct mathematical model of the problem environment, we find someone who knows how to solve the problem, and develop a mathematical model of their solution process. The author contends that some combination of analytic and cognitive solutions represent an appropriate strategy for the Epistemic Challenge.

2.3 Cognitive Solutions

When engineering cognitive solutions, the author subscribes to the philosophical doctrine of functionalism [9] which attributes identity on the basis of functional equivalence. This allows us to think of a bird and a jet fighter as both being able to fly, even though the underlying mechanisms differ beyond some level of scrutiny. Functionalism equally allows us to view the analytic search strategies of Deep Blue as reasoning about chess moves. Here cognitive solutions are likewise interpreted as functionalist equivalents of activity undertaken by people. They are not professing to mirror neurological processes.

With analytically difficult domains in which human expertise is available, an appropriate strategy is to model the cognitive behaviour of people. If we are to repeatedly model human solutions for machine information fusion, then it is desirable to have:

1) a general cognitive modelling framework;
2) a means of capturing an individual’s cognitive behaviour within that general framework; and
3) a means of automating an individual’s captured cognitive behaviour within a machine.

The following three sections explore these three issues.

3 Cognitive Model

3.1 Propositional Attitudes

Propositional attitude expressions are expressions of the form <subject> <attitude> that <propositional expression>. They are the means by which each of us relates our own mental behaviour to others, and the means by which each of us communicates the mental activity we conclude from the behaviour of others. Propositional attitudes are alleged mental states characterised by propositional attitude expressions, where: <subject> denotes the individual whose mental state is being characterised (e.g. Fred believes that it is raining); <propositional expression> is describing some propositional claim about the world (e.g. Fred believes that it is raining, Tom believes that it is raining); and <attitude> is expressing the subject's dispositional attitude toward that claim about the world (e.g. Fred believes that it is raining, Fred believes that the sky is blue); and <propositional expression> is expressing the subject's dispositional attitude toward that claim about the world (e.g. Fred believes that it is raining, Fred hopes that it is raining).

The cognitive modelling framework presented in this paper is called ATTITUDE because it is based on propositional attitude expressions. This does not necessarily endorse the realist position that propositional attitudes genuinely exist, but it does at least endorse an
instrumentalist claim that propositional attitudes are a sufficient engineering tool.

3.2 Individuals

Within the ATTITUDE model, a society \( S = \{X_1, \ldots, X_n\} \) is a collection of individuals \( X_i \). Each individual \( X \in S \) is an evolving process, and so is modelled as a function of time \( X : \text{Time} \rightarrow \text{State} \). Hereafter \( X_t \) denotes the state \( X(t) \) of individual \( X \) at time \( t \). The state \( X_t \) is in turn modelled in terms of both memory and control, and consequently \( X_t = \text{Memory}_{X_t} \cup \text{Control}_{X_t} \). Modelling \( \text{Memory}_{X_t} \) in terms of both working memory \( \text{WM}_{X_t} \) and long-term memory \( \text{LTM}_{X_t} \), produces the equation \( X_t = (\text{WM}_{X_t} \cup \text{LTM}_{X_t}) \cup \text{Control}_{X_t} \). Thus, under the ATTITUDE model,

\[
X_t = (\text{WM}_{X_t} \cup \text{LTM}_{X_t}) \cup \text{Control}_{X_t} \quad (1).
\]

3.3 Working Memory

Working memory holds the individual’s state of awareness, volition, pending mental actions and interactions toward its environment.

\[ \text{WM}_{X_t} = \text{Awareness}_{X_t} \cup \text{Volition}_{X_t} \]

\[ \cup \text{Action}_{X_t} \cup \text{Interaction}_{X_t} \quad (2). \]

3.3.1 Awareness

Under the ATTITUDE model, the author characterises an individual’s awareness in terms of beliefs, expectations and anticipations.

\[ \text{Awareness}_{X_t} = \text{Beliefs}_{X_t} \cup \text{Expectations}_{X_t} \]

\[ \cup \text{Anticipations}_{X_t} \quad (3). \]

Each expectation is modelled by a sentence of the form \( X \) expects that \( \alpha \) by \( t \), where: \( m \in \text{Id}_e \) is a unique identifier for each current expectation; \( X \) refers to \( X \in S \); \( \alpha \in \text{Expr} \) is an expression; and \( t \in \text{Time} \). It indicates \( X \)’s expectation that \( \alpha \) will be the case by \( t \). Each anticipation is modelled by a sentence of the form \( X \) anticipates that \( \alpha \) with \( \beta \), where: \( m \in \text{Id}_a \) is a unique identifier for each current anticipation; \( X \) refers to \( X \in S \); and \( \alpha, \beta \in \text{Expr} \) are expressions. It indicates \( X \)’s anticipation that \( \alpha \) will occur and that \( X \) will respond by intending that \( \beta \) should occur.

Beliefs are modelled in terms of conditional, unconditional and inference beliefs, \textit{id est}:

\[ \text{Beliefs}_{X_t} = \text{Conditional}_{X_t} \cup \text{Unconditional}_{X_t} \]

\[ \cup \text{Inferences}_{X_t} \quad (4). \]

Each unconditional belief is modelled by a sentence of the form \( X \) believes that \( \xi \) in event \( E \) with priority \( r \) and probability \( p \), where: \( m \in \text{Id}_b \) is a unique identifier for each unique belief; \( X \) refers to \( X \in S \); \( \xi \) is an expression \( \alpha \) or a negated expression not \( \beta \), for \( \alpha \in \text{Expr} \); \( E \in P(\text{Beliefs}_{X_t}) \) is a partitioned subset of beliefs associated with an event (\( P \) denotes the powerset defined by \( P(x) = \{u \mid u \subseteq x\} \)); \( r \in [0,1] \) is the belief’s priority; and \( p \in [0,1] \) is the probability associated with the belief. It indicates that \( X \) believes that \( \xi \) is the case in event \( E \) with probability \( p \).

Each conditional belief is modelled by a sentence of the form \( m. X \) believes that \( \xi \) if \( \zeta_1 \) and … and \( \zeta_j \) in event \( E \) with priority \( r \) and probability \( p \), where: \( m, X, \xi, E \) and \( r \) are as for unconditional beliefs; \( \zeta_m \) is an expression \( \alpha \), a strongly negated expression that not \( \alpha \), or a weakly negated expression not that \( \alpha \); and \( p : P(\{\zeta_1, \ldots, \zeta_j\}) \rightarrow [0,1] \) is a conditional probability association for the conditional belief. It indicates that \( X \) believes that \( \xi \) follows from \( \zeta_1 \) and … and \( \zeta_j \) in event \( E \) with the conditional probabilities of \( p \).

Each inference is modelled by a sentence of the form \( \alpha \) inferring \( \alpha \) in event \( E \) with deduction \( d \) with options \( \lambda \), where: \( m, X, \alpha, E \) and \( \alpha \) are as before; \( d \) references a deductive structure; and \( \lambda \) identifies unconditional beliefs potentially matching \( \alpha \) or conditional beliefs with consequent potentially matching \( \alpha \).

3.3.2 Volition

Under the ATTITUDE model, the author characterises an individual’s volitions in terms of intentions and desires.

\[ \text{Volition}_{X_t} = \text{Intentions}_{X_t} \cup \text{Desires}_{X_t} \quad (5). \]

Each intention is an independent goal modelled by a sentence of the form \( m. X \) intends that \( \alpha \) with subgoals \( G \) and priority \( r \), where: \( m \in \text{Id}_i \); \( X \) refers to \( X \in S \); \( \alpha \in \text{Expr} \) is an expression; and \( r \in [0,1] \) is the intention’s priority. It indicates that \( X \) intends \( \alpha \) to be the case and has the subgoals \( G \) in place to achieve \( \alpha \).

Desires are dependent goals with higher intentions that form hierarchical structures and derive from past experiences stored as routines (see section 3.4). Each desire is modelled by a sentence of the form \( m. X \) desires that \( \sigma \) from routine element \( q \) with context \( f \) for superior goal \( g \) with intention \( i \) and with subgoals \( G \), while associated with task \( t \) and expiry \( e \), where: \( m \in \text{Id}_d \); \( X \) refers to \( X \in S \); \( \sigma \in \text{Expr} \) is the propositional attitude expression of routine element \( q \in \text{Id}_r \); \( f \in \text{Var}_{\text{Expr}} \) provides context by assigning expression values to variables; \( g \in \text{Id}_g \); \( i \in \text{Id}_i \); \( G \in P(\text{Id}_g) \); \( t \in \text{Id}_t \) and \( e \in \text{Id}_e \). Desires are explained in section 3.4.

3.3.3 Action

Under the ATTITUDE model, the author characterises an individual’s pending actions through tasks and expirations.

\[ \text{Action}_{X_t} = \text{Tasks}_{X_t} \cup \text{Expirations}_{X_t} \quad (6). \]

Each task is a scheduled mental action modelled by a sentence of the form \( m. X \) tasks that \( \omega \) for \( Y \) has priority \( r \), where: \( m \in \text{Id}_a \); \( X \) refers to \( X \in S \); \( Y \) refers to a collection \( Y \in P(S) \); \( \omega \in \text{Id}_o \cup \text{Id}_c \) references a desire where \( X \) is \( Y \), and references a message where \( X \) is not \( Y \); and \( r \in [0,1] \) is the task’s priority. It specifies a scheduled action to attempt to satisfy a desire or attend to a message. Each expiration is a scheduled expiration expiry modelled by a sentence of the form \( m. X \) expires action \( \omega \).
by time $t$, where $m \in Id_{\pi}$; $X$ refers to $X \in S$; $\omega \in Id_{\omega}$; and $t \in \text{Time}$ is the expiry time.

3.3.4 Interactions

Under the ATTITUDE model, communication between individuals is modelled separately from more general sensing and effecting of the environment.

$$\text{Interaction}_{X,t} = \text{Sensation}_{X,t} \cup \text{Effect}_{X,t} \cup \text{Communication}_{X,t}$$  \hspace{1cm} (7)

Each sensation is a buffered processed sensory input modelled by a sentence of the form $m$. $X$ senses that $\alpha$ with priority $r$ through sensor $s$, where: $m \in Id_{\pi}$; $X$ refers to $X \in S$; $\alpha \in \text{Expr}$ is an expression; and $r \in [0,1]$ is the sensation’s priority.

Each effect is a buffered effecter output instruction modelled by a sentence of the form $m$. $X$ effects that $\alpha$ with priority $r$ through effector $e$, where: $m \in Id_{\pi}$; $X$ refers to $X \in S$; $\alpha \in \text{Expr}$ is an expression; and $r \in [0,1]$ is the effect’s priority.

Each communication is a buffered message modelled by a sentence of the form $m$. $X$ informs $Y$ that $\Psi \alpha$ with priority $r$ and index $m_2 \theta$, where: $m \in Id_{\pi}$; $X$ refers to $X \in S$; $\alpha \in \text{Expr}$ is an expression; and $r \in [0,1]$ is the message’s priority; $\theta \in$ [is requested, succeeded, failed] indicates whether the message is a request or the result of a request; and $m_2$ is the message index, which is $m_1$ in the case of a request.

3.4 Long Term Memory

The author contends that behaviour often derives from the execution of recipes for mental behaviour formed from past experiences. These repeatable recipes for mental behaviour are called routines in the ATTITUDE model, as the term conveniently marries its everyday sense of "customary" with its computer science sense of "procedural". Routines are both the memories of past experience and the guides for future experience.

Each routine $R \in LTM_{X,t}$ is composed of a goal expression $g \in \text{Expr}$ and a network of instructions designed to achieve goal $g$. The network is modelled as a set of propositional attitude nodes, in which each node has the form $m$. $<\sigma, s, t>$, where: $m \in Id_{\pi}$; $\sigma \in \text{Expr}$ is a propositional attitude instruction; and $s, t \in Id_{\omega}$. A propositional attitude instruction is the instructional counterpart of a propositional attitude observation. For example, the counterpart to the propositional attitude observation Fred believes that it is raining is the propositional attitude instruction Fred believe it is raining, instructing individual Fred to believe that it is raining.

When each propositional attitude instruction $\sigma$ succeeds or fails, control is directed to the nodes identified by the $s$ or $t$ values respectively. The cognitive routine networks can be visualised as success-fail (SF) diagrams by linking propositional attitude instructions through $S$ and $F$ edges. For example, the set of propositional attitude nodes  

[Figure 3. Routine network diagram]

A routine will be selected from long-term memory when its goal expression matches a current goal. The match establishes a context $f$ that maps variables from expressions to values. $f(\alpha)$ is the expression formed from expression $\alpha$ after each variable $?v$ in $\alpha$ is replaced by its contextual value $f(?v)$. When a routine is invoked, a desire is created for each initial node of the routine’s network. The desire references: its corresponding routine node, together with the context for the node; the goal that activated the routine, together with its intention; a scheduled task to attempt to satisfy the desire; and any goals that subsequently come to depend upon the desire. When a task is selected for execution, the propositional attitude instruction referenced by its desire is performed relative to the desire’s context. The outcome is usually success or failure; a modification of working memory; a modification of the context, and a redirection of the desire (with its new context) to a subsequent routine node, determined by its success or failure. Desires contextually transition routines in this way, and collectively succeed when a terminal node succeeds.

ATTITUDE includes propositional attitude instructions involving awareness, volition, action, interaction, and control. The following ATTITUDE propositional attitude instructions are subsequently referenced in this paper:

- $X \alpha \beta \xi \in E \Gamma$ succeeds if subject $X$ acquires a new belief $f(\alpha \beta \xi)$ with priority $r$ in event $E$ within $\text{LTM}_{X,t}$.
- $X \alpha \beta \xi \in E \Gamma$ succeeds if the unification pattern match of $\alpha \beta \xi \Gamma$ succeeds if the inference structure succeeds if subject $X$ has a belief in $\text{LTM}_{X,t}$ matching $f(\alpha \beta \xi \Gamma)$, establishes a new inference structure $\Gamma$ in $\text{LTM}_{X,t}$, and sets the subsequent context $f^*$ to include the variable assignments resulting from the match.
- $\text{ask} \Gamma \alpha \beta \xi \Gamma$ succeeds if the inference structure $\Gamma$ determines another belief that matches the query associated with $\Gamma$, modifies the inference structure $\Gamma$ accordingly, and sets the new context $f^*$ to include the variable assignments resulting from the match.

$\text{not ask} \Gamma \alpha \beta \xi \Gamma$ succeeds if the inference $\Gamma$ is removed.
- $X \alpha \beta \xi \in E \Gamma$ succeeds if subject $X$ has one (or more) routines $R \in \text{LTM}_{X,t}$ with a goal expression $g$ matching $f(\alpha \beta \xi)$, successfully transitions routine $R$ commencing with context $f$, and uses the final context $f^*$ to output the resulting variable assignments.
- $X \alpha \beta \xi \in E \Gamma$ succeeds displaying $f(\alpha \beta \xi)$.
- $\text{match}(\alpha) \beta \xi \Gamma$ succeeds if the unification pattern match of $f(\alpha \beta \xi)$ and $f(\beta)$ succeeds and sets the subsequent context $f^*$ to include the resulting variable assignments. For example, match((?x big) (4 ?y)) will succeed assigning ?x to 4 and ?y to big.
3.5 Control

Control is the process that drives activity. It repeatedly checks expiration elements; selects the next highest priority task; performs the contextual propositional attitude instruction associated with that task; and then redirects according to the success or failure of that task.

4 Capturing Cognitive Routines

4.1 Capture Process

The ATTITUDE approach to cognitive routine capture involves applying the following steps to a collection of representative cases.

1. **Speech to Text**: the expert dialogues as they are dealing with each case, with their utterances being recorded and converted to text. Instrumentation of their movements can also be undertaken if required.

2. **Corrected Text**: the generated text is then corrected to remove automated speech to text conversion errors and to capture the appropriate nomenclature.

3. **Text Analysis**: the corrected text is then analysed to expose the goal being attempted and the steps by which the attempt was made, noting the success and failure of each step.

4. **Attitude Analysis**: each of the steps identified in the text analysis are represented as a propositional attitude instruction. This involves settling on an ontological framework for the case domain and the classification of each step in terms of one of the ATTITUDE attitudes.

5. **Generalised Analysis**: the goal and each of the Attitude Analysis steps has elements of it generalised through parameterisation. The expert needs to assist in the choice of parameterisation.

6. **Superimposition Analysis**: a SF network resulting from the Generalised Analysis is developed and superimposed cumulatively onto the results of the Superimposition Analysis of previous cases.

4.2 Cognitive Capture Example

To illustrate the ATTITUDE approach to cognitive routine capture, a simple problem is required. The selected problem was to capture experience at finding the linear factors of quadratic equations. The problem is clearly not of mainstream concern to the information fusion community, but it nonetheless affords three explanatory advantages.

Firstly, the quadratic factoring problem facilitates exposition of the proposed approach toward the Epistemic Challenge, without having to confront the considerable complexities of the Semantic Challenge. Most problems of interest to the information fusion community cannot be explored without a rich semantic framework encompassing time, space, objects, intents, et cetera. The semantic and epistemic challenges are inextricably linked.

Secondly, the quadratic factor problem contrasts the analytic and cognitive approaches. The quadratic root problem is simple enough to admit a closed form analytic solution. A general quadratic \( ax^2 + bx + c \) will equal zero when \( x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \), and this can be used to produce linear factors. This is how we would usually solve the problem analytically by computer, and it is characteristic of the analytic approach of the sensor fusion community. But it is not how the author cognitively factorises \( x^2 + 3x + 2 \). Of course, the cognitive alternative is particularly valuable when it can be applied to problems for which analytic solutions are difficult to obtain.

Thirdly, the quadratic factor problem is simple enough to avoid the need for sensors and effectors and requires only one individual in the problem solving process. Because the subject “I” in the following routines could just as easily have been the name of another individual, the problem allows the reader to appreciate how the ATTITUDE framework can model distributed cognitive behaviour without having to consider a more complex problem involving multiple individuals.

A collection of case studies was required to extract quadratic routines. The author used the following set of questions from [10] as case studies.

\[
\begin{align*}
x^2 - 5x + 6 &= 0; & y^2 + y - 20 &= 0; & w^2 - 64 &= 0; \\
4a^2 - 28a &= 0; & 4x^2 - 1 &= 0; & 9b^2 - 3b &= 0; \\
c^2 + 6c + 8 &= 0; & d^2 - 5d - 24 &= 0; & 3p^2 + 20p - 7 &= 0; \\
2r^2 + 3r - 5 &= 0; & t^2 + 8t - 20 &= 0; & -2x^4 + 9x - 7 &= 0.
\end{align*}
\]

The following reports the six step capture process being applied to the first two cases, beginning with the automated speech recognition record of the author’s verbalised solution process.

\[
x^2 - 5x + 6 = 0
\]

**ST**: “The first question is \( x^2 - 5x + 6 = 0 \). I want factors of six. Three and to give six but I need their son to be minus 5. Minus 2 and minus 3 will multiply to give six and add to give minus 5. So if I have eggs minus 2 times next minus 3 then I will get eggs squared minus three acts minus 2 ex plus 6 which gives ex squared minus five ex plus 6 as required”.

**CT**: The first question is \( x^2 - 5x + 6 = 0 \). I want factors of 6. 3 and 2 give 6, but I need their sum to be -5. -2 and -3 will multiply to give 6 and add to give -5. So if I have \( x = 2 \) . \( x = 3 \) then I will get \( x^2 - 3x - 2x + 6 \), which gives \( x^2 - 5x + 6 \) as required.

**TA**: **Goal**: In this case, the goal is to determine values for \( m \) and \( n \) so that the quadratic equation \( x^2 - 5x + 6 = 0 \) can be expressed as a linear product \( (x \pm m)(x \pm n) = 0 \).

**Steps**: The first step was to look at the 6 term. The fact that \( 2 \cdot 3 = 6 \) was retrieved from memory. **Success**.

The second step was to require these factors to sum to -5. Clearly 2 and 3 don’t. **Failure**.
In the third step I recalled that \(-2 \cdot -3 = 6\) too. Success.
In a fourth step I noted that \(-2 + -3 = -5\). Success.
In a fifth step I considered \((x - 2) \cdot (x - 3)\) and checked the proposed solution by evaluating \(x \cdot x + x \cdot -3 + -2 \cdot x + -2 \cdot -3\) and matching it against \(x^2 - 5x + 6\). Success.
The final step is the acceptance of the solution \((x - 2) \cdot (x - 3)\). Success.

**AA: Goal:** In this case, the goal is \((\text{factorise } x^2 - 5 \cdot x + 6)\) into \((x \text{ ?op1 ?m} \cdot x \text{ ?op2 ?n})\).

**Steps:** I ask if believe (times ?u ?v is ?c) in ?times_tables with for_factors. Success.
match \((+ ?u ?v) \cdot (- ?b))\). Success.
match \((x^2 + (+ ?p ?q) \cdot x + (\times ?p ?q)) \cdot (x^2 + (- ?b) \cdot x + ?c))\).
Success.
match \(((x \text{ ?op1 ?m} \cdot x \text{ ?op2 ?n})) \cdot (x - (\text{abs ?p} \cdot x - (\text{abs ?q})))\).
Success.
match \(((x \cdot x + x \cdot -3 + -4 \cdot x + -4 \cdot -5 \cdot x + -20)\).
Success.
This gives \(x \cdot x + x \cdot -4 \cdot x + -4 \cdot 5 \cdot x + -20\) which reduces to \(x^2 + x - 20\) as desired.

**TA: Goal:** In this case, the goal is to determine values for m and n so that the quadratic equation \(y^2 + y - 20 = 0\) can be expressed as a linear equation product \((y \pm m) \cdot (y \pm n)\).

**Steps:** I first contemplated \(-20\), which means that I had reformatted the quadratic \(y^2 + y - 20\) as the quadratic \(y^2 + y + (-20)\). Success.
I then considered the \(-20\) term. \(4 \cdot 5 = 20\) was retrieved from memory. Success.
To get a product of \(-20\), one of these two factors needs to be negative. I selected \(+4\) and \(-5\) and, in a more direct approach than with the first case, I checked the proposed solution immediately by evaluating \(x \cdot x + x \cdot -3 \cdot x + -3\cdot y + y + (-4.5)\) and matching it against \(y^2 + y + 20\). This match failed. (Interestingly I had inadvertently used \(x\) rather than \(y\) in framing the solution. Clearly the variable was regarded as parametric.) Failure.

In the next step I considered the alternative assignment with \(-4\) and \(+5\), again evaluated \(x \cdot x + x \cdot 5 \cdot x + -4 \cdot x + -4\cdot 5\) and successfully matched it against \(y^2 + y - 20\). However, the match only holds if \(y^2 + y - 20\) is understood as \(y^2 + 1 \cdot y + -20\). Success.
The final step is acceptance of the solution \((x - 4) \cdot (x + 5)\). Success.

**AA:** I express this as steps from three routines: the first to reformat the quadratic; the second to handle the \(1\) term; and the third to undertake the factorisation.

**Goal:** \((\text{reformat } (y^2 + y - 20) \text{ as } (y^2 + y + -20))\)

**Steps:** approve. Success.

**Goal:** \((\text{factorise } (y^2 + y + -20) \text{ into } (y \text{ ?op1 ?m} \cdot y \text{ ?op2 ?n}))\)

**Steps:** I desire \((\text{factorise } (y^2 + 1 \cdot y + -20) \text{ into } (y \text{ ?op1 ?m} \cdot y \text{ ?op2 ?n}))\).

**Goal:** \((\text{factorise } (y^2 + 1 \cdot y + -20) \text{ into } (y \text{ ?op1 ?m} \cdot y \text{ ?op2 ?n}))\)

**Steps:** I ask if believe (times ?u ?v is 20) in ?times_tables with for_factors. Success.
match \(((y^2 + (+ ?u (- ?v))) \cdot y + (\times ?u (- ?v))) \cdot (y^2 + 1 \cdot y + -20))\).
Failure.
match \(((y^2 + (+ (- ?u) ?v) \cdot y + (\times (- ?u) ?v)) \cdot (y^2 + 1 \cdot y + -20))\).
Success.

**GA:** I express this as steps from three generalised routines.

**Goal:** \((\text{reformat } (y^2 + y \text{ ?op ?c}) \text{ as } (y^2 + y + ?sign_c))\).
Steps: match ([(?op (- ?c)) (- ?sign_c)]). Success.
Goal: (factorise (?y^2 + ?y + ?c) into (?y ?op1 ?m) (?y ?op2 ?n)).
Steps: I desire (factorise (?y^2 + 1 ?y + ?c) into (?y ?op1 ?m) (?y ?op2 ?n)) Success.
Goal: (factorise (?y^2 + ?b ?y + ?c) into (?y ?op1 ?m) (?y ?op2 ?n)).
Steps: I ask if believe (times ?u ?v) is (abs ?c)) in ?times_tables with for_factors. Success.
match (y^2 + (+ ?u (- ?v))) y + (x^2 (- ?u)) (y^2 + ?b y + ?c)). Failure.
mismatch (y ?op1 ?m) (y ?op2 ?n)) (y - ?u) (y + ?v)).
Success.
SA: The superimposition involves a few decisions: (a) a generalised quadratic x^2 ± bx ± c will be represented by (x^2 ?op1 ?b x ?op2 ?c); (b) test to distinguish the first and second cases on the basis of the sign of the ?c term after reformatting; (c) apply the more efficient pattern matching of the second case to the first case; (d) parameterise the variable to handle the x in case 1 and the y in case 2. Quadratic x^2 ± bx ± c will be represented by (x^2 ?op1 ?b x ?op2 ?c). Cases 1 and 2 yielded the following routines.

Figure 5. (reformat (?x^2 ?op1 ?b ?x ?op2 ?c) as (?x^2 + ?sign_b ?x + ?sign_c))

Figure 6. (factorise (?x^2 ?op1 ?b ?x ?op2 ?c) into (?x ?op1 ?m) (?x ?op2 ?n))

Figure 7. (reformat (?x^2 ?op1 ?b ?x ?op2 ?c) as (?x^2 + ?sign_b ?x + ?sign_c))

Figure 8. (factor (?x^2 + ?b ?x + ?c) into (?x ?op1 ?m) (?x ?op2 ?n))

By continuing the process, a collection of cognitive routines is assembled from the cases.

5 Computing Cognitive Routines
The final consideration is computation of the captured cognitive routines. This is done through the ATTITUDE multi-agent reasoning system. The ATTITUDE system is so named because it codes in terms of ATTITUDE's propositional attitudes expressions, which significantly eases the burden of translation from capture to code.

An individual quadratic_factoriser agent was coded in ATTITUDE to express the captured routines. The cases
\[ x^2 - 5x + 6 = 0; \quad y^2 + y - 20 = 0; \]
\[ c^2 + 6c + 8 = 0; \quad d^2 - 5d - 24 = 0 \]
contributed to the formation of particular superimposed routines with reformat, factorise and factor goals. The superimposed factor routine appears in Figure 9.

Figure 9. (factor (?x^2 + ?b ?x + ?c) into (?x ?op1 ?m) (?x ?op2 ?n))

To code this routine it is necessary to express the SF diagram in terms of control constructs. Figure 10 describes three of the control constructs available in ATTITUDE.

\[^{(\alpha_1 \ldots \alpha_l) succeeds if \alpha_1 succeeds, and then \ldots, and then \alpha_l succeeds; and fails once one \alpha_i fails. \]
\[](\alpha_1 \ldots \alpha_l) succeeds if \alpha_1 succeeds, and \alpha_2 succeeds, or \ldots or \alpha_l succeeds, or \ldots or \alpha_1 \ldots \alpha_{l-1} fail and then \alpha_l succeeds; and fails if each \alpha_1 \ldots \alpha_l fail in turn.

\[^{(\alpha) succeeds if \alpha is repeatedly attempted and eventually fails; and cannot fail because it will not succeed only if \alpha succeeds infinitely.\]

Figure 10. Control constructs

ATTITUDE code for the routine of Figure 9 follows.

routine (factor (?x squared plus ?b ?x plus ?c) into (?x ?op1 ?m) (?x ?op2 ?n)) is
\[(^\wedge \text{I ask if believe (times ?u ?v) is (abs ?c)}) \text{ in } \text{?times_tables with for_factors}
\]
\[\{\wedge \text{match (> ?c 0) true}
\]
\[\{\wedge (^\wedge \text{match (> (abs (- (+ ?u ?v) (abs ?b)) (abs ?c))) true ask again for_factors))
\]
\[\text{not ask for_factors match (+ ?u ?v) (abs ?b)}\]
\[\{\wedge \text{match (> ?b 0) true}
\]
\[\{\wedge \text{match ((?x squared plus (+ ?u ?v) ?x plus (x ?u ?v)) (x squared plus ?b ?x plus ?c))}
\]
\[\text{factorise (y 2 + y + ?c) into (y ?op1 ?m) (y ?op2 ?n))
\]
\[\{\wedge \text{match ((?x squared plus (+ (- ?u) (- ?v)) ?x plus (x (- ?u) ?v))}
\]
\[\text{match ((?x ?op1 ?m) (?x ?op2 ?n)) (y squared plus ?b ?x plus ?c))}
\]
\[\text{match ((?x ?op1 ?m) (?x ?op2 ?n))}
\]
(((?x minus ?u) (?x minus ?v)))
-disapprove))
(^ (* (^ match ((> (abs (- (abs (- ?u ?v)) (abs ?b))) 0)
true) ask again for_factors))
not ask for_factors match ((abs (- ?u ?v)) (abs ?b))
l1 (^ match ((> ?b 0) true)
| (^ match ((> ?u ?v) true)
| (^ match ((> ?u ?v) true)
| (^ match (((?x ?op1 ?m) (?x ?op2 ?n))
((?x plus ?u) (?x minus ?v))))
-disapprove))
(^ match (((?x squared plus (+ ?u (- 0 ?v)) ?x
plus (x ?u (- 0 ?v)))
(?'x squared plus ?b ?x plus ?c))
match (((?x ?op1 ?m) (?x ?op2 ?n))
((?x plus ?u) (?x minus ?v))))
-disapprove))
(^ match (((?x squared plus (+ (- 0 ?u) ?v) ?x
plus (x (- 0 ?u) ?v)))
(?'x squared plus ?b ?x plus ?c))
match (((?x ?op1 ?m) (?x ?op2 ?n))
((?x minus ?u) (?x minus ?v))))
-disapprove))
(^ match (((?x squared plus (+ 0 ?u) ?v) ?x
plus (x (- 0 ?u) ?v)))
(?'x squared plus ?b ?x plus ?c))
match (((?x ?op1 ?m) (?x ?op2 ?n))
((?x minus ?u) (?x minus ?v))))
-disapprove))
(^ match (((?x squared plus (+ ?u (- 0 ?v)) ?x
plus (x ?u (- 0 ?v)))
(?'x squared plus ?b ?x plus ?c))
match (((?x ?op1 ?m) (?x ?op2 ?n))
((?x minus ?u) (?x minus ?v))))
-disapprove)));

Coding this with related routines, times table knowledge,
and our 4 cases as propositional attitude instructions like
I desire (factorise (u squared minus 5 u plus 6)
into (u ?op1 ?a) (u ?op2 ?b))
I show (expression factorises (u squared minus 5 u plus 6)
into (u ?op1 ?a) (u ?op2 ?b))
in the quadratic_factor individual, produces the output
factorises ( u squared minus 5 u plus 6 )
into ( u minus 2 ) ( u minus 3 )
factorises ( y squared plus y minus 20 )
into ( y minus 4 ) ( y plus 5 )
factorises ( c squared plus 6 c plus 8 )
into ( c plus 2 ) ( c plus 4 )
factorises ( d squared minus 5 d minus 24 )
into ( d minus 8 ) ( d plus 3 ).

6 Conclusions

The paradigm shift from sensor fusion to information
fusion requires a symbolic approach to information fusion.
This in turn induces a Semantic Challenge and an
Epistemic Challenge. This paper argues that some
combination of cognitive and analytic solutions represent
an appropriate strategy for the Epistemic Challenge. To
that end the paper presents a general cognitive approach
involving: a cognitive model; a process for cognitive
capture; and a computational framework for automating
the cognitive routines. Analytical techniques can then be
combined with these routines. For example, by viewing
cognitive routines as mechanisms for transitioning one’s
awareness of the world (often by transforming the world),
the optimal selection of cognitive routines over time can
be formulated as a dynamic programming problem,
minimise \( \sum_{t \in T} \text{mismatch}(V_{t,k}, A_k) \) (8)
in which: mismatch is a proposition set distance measure;
\( V_{t,k} \) specifies the volitions held at time \( t \) about some future
time \( k \); and \( A_k \) is the awareness expected at future time \( k \),
given some selection of cognitive routines.

References