A Reference Model for Data Fusion Systems

Mieczyslaw M. Kokar
Dept. Electrical and Computer Engineering
Northeastern University
360 Huntington Avenue
Boston, MA 02115
USA
kokar@coe.neu.edu

Mark D. Bedworth
School of Computing
University of Central England
Perry Barr
Birmingham, B42 2SU
UK
mark.bedworth@datafusion.clara.co.uk

Carl B. Frankel
Organizational Measurement and Engineering
785 Burnett Avenue, No 2
San Francisco
CA 94131-1417
USA
carlf@omel.com

ABSTRACT

In this paper we offer an overview of design principles and propose a fusion process reference model that provides guidance for the design of data fusion systems. We incorporate a formal method approach to fusion system design and show the rôle of the psychology of the human / computer interface in the system design process. Data fusion is a complex, multi-faceted field that has evolved from a number of different disciplines. This disparate nature has lead to a largely bottom-up approach to data fusion system design where the components are constructed first and the system-level issues addressed afterwards. The result is an *ad hoc*, prototype driven philosophy which, we content, is neither efficient nor effective. We believe that design of data fusion systems needs to be given proper consideration, with a top-down approach that addresses system-level constraints first, thereby offering the possibility of re-usable, abstract structures. We offer an object-centred model of data fusion together with practical tools for studying and refining the model so that it can be useful in designing *real* data fusion systems.

Keywords: Information fusion, process model, object orientation, formal methods, Petri-nets, cognitive psychology.

1. BACKGROUND

"When all is said and done about data fusion, how come there's so much more said than done?" [1]

The observation above is sadly all too true and strikes a resonant chord with many a customer of data fusion systems. Since data fusion went "international" in the final decade of the last millennium, the world seems to have been steadily filling up with prototype data fusion systems and an (almost) equal quantity of disappointed customers. So what has been going wrong – why are we so naïve at turning obviously successful techniques into truly useful (and usable) systems?[2] The authors believe that the answer lies in the way most of us are viewing data fusion. By concentrating on fusion as a means of operating on data, we make the error of thinking in terms of data fusion systems[†] rather than systems with a data fusion capability.

Research is leveraged through re-use, and in the multi-disciplined field of data fusion re-usability is required, not only of code, but also of architectures, design specifications and operating principles. In this paper we take apart conventional notions of fusion process modelling and re-invent it from an object-centred, multiple perspective viewpoint. We do this

---

[†] Despite our reticence about the appellation *"data fusion system"* we shall continue to use the term herein for the grammatical convenience it offers the reader. Wherever it appears the reader should think of it as describing a system that possesses a fusion capability rather than implying a data fusion system in its own right.

specifically with the intention of developing re-usable concepts and methodologies for designing and developing a data fusion capability at the system level. From a system design viewpoint, data fusion is not about data but about structures.[12,10] By combining theories and models rather than data it is possible to reason about the system before the system is actually constructed. We make no claims that we have addressed all of the problems, rather that we have defined a crucial, but neglected, requirement and made the first tentative steps towards its solution.

2. INTRODUCTION

> **Design** (di'zain) *verb*: To (skilfully) work out the structure or form of something,
> to plan or invent it, as intended for a specific purpose.

To work out the structure of a data fusion system and ensure that it is fit for the intended purpose, at least three issues should be addressed:

- A process model should be constructed that is simple enough to allow predictive analysis yet rich enough to admit a variety of complex designs. The model should admit a process of gradual, piece-wise refinement which eventually leads to the deployable system but at the same time should not be overly constraining (and thus giving the designer enough flexibility for choosing design solutions)

- A framework for specifying tasks (goals and solutions) should be developed which permits achievability to be assessed without the need to implement the system. This necessitates a means of representing the state of available knowledge about the world, the information sources and the effectors (but not necessarily a representation of the knowledge itself)

- A theory for the dynamics of the information and of the information processors should be studied. The latter should incorporate both humans and machines and allow for the decision making process to be modelled as a team activity

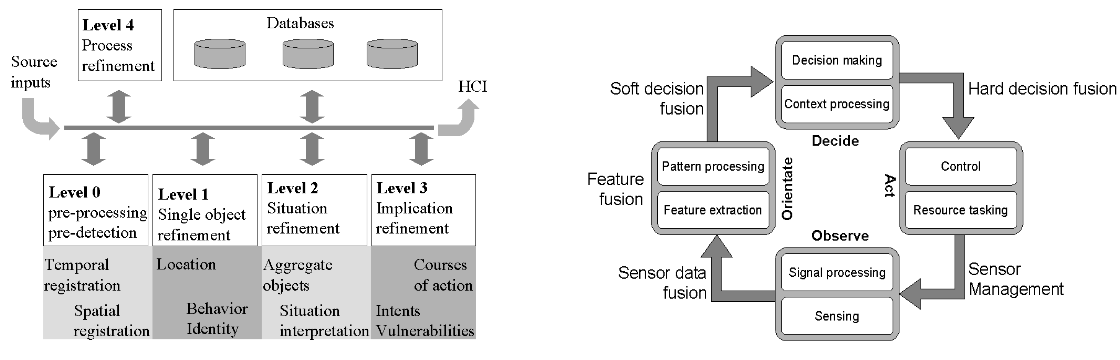
We propose a fusion reference model that addresses these issues in the following way:

- A prescriptive but tractable object-relationship model
- A formal method for analysing the relationships between objects
- A mathematical tool for simulating the information flow at the system level
- An appreciation of the psychological factors in the human components

By applying these concepts to data fusion substructures, and the way they may be composed, the important distinction between design and implementation may be clarified. It is our intention that this will eventually lead to a set of standard data fusion substructures that have known operating behaviour, both individually and in context. Our aim is to increase the awareness of system design considerations in data fusion and to start the transition of this endeavour from an art to a science. In this final respect we believe that there is still far to go.

4. PROCESS MODEL

A process model is a description of a set of processes and the relationships between them. The set of processes such determined should be constructed before the system may be regarded as fully operational. As such it highlights the component functions which the system has but makes no statement regarding their software implementation or physical instantiation. Existing data fusion models (such as JDL[3] and Omnibus[4] for example, see figure 1) are still largely descriptive models and thus cover a large number of designs that are it should be able to rule out, even before introducing the constraints imposed by the specific application.



(a) *(b)*

Figure 1a) The updated JDL data fusion process model and (1b) the Omnibus model introduced in 1999.

4.1 Information Centred

In the fusion model proposed by the US Joint Directors of Laboratories Data Fusion Sub-Group[3], the Dasarathy model[5] and the Waterfall model[6] the data fusion process is divided according to the abstraction of the information being fused. There is therefore no need to make sequencing of functions or processors explicit. Typically the levels of abstraction include:

Sensor data – such as scalar measurements, waveforms or images

Signals – which are the result of some elementary signal processing, alignment or registration activities

Features – which capture the relevant characteristics of the signal and represents the lowest level at which desired capabilities affect the information being stored

Object state estimates - which is concerned with the estimation and prediction of continuous (*e.g.* spatial or kinematic) or discrete (*e.g.* behaviour or identity) states of objects

Situation estimates – which introduces context by examining the relations among entities, aggregating objects into meta-objects and placing interpretations on the situation

Planning decisions – implication and possible courses of action are analysed in light of the current situation

4.2 Function Centred

While the models referred to in the previous section are centred on the level of abstraction of the information, the intelligence cycle[7], the Boyd control loop[8] and the Omnibus model[4] are mainly organised from a functional viewpoint. In this case the sequence of functions that are to be performed are made explicit. Each of these models uses a four-stage model:

Feeding – observing or collecting information and passing it on

Informing – collating and orientating the information to increase its relevance

Directing – by evaluating choices and making decisions

Managing – implementing those decisions

Implied by this structure are the agents that perform the functions (actors, perceivers, directors and managers). The merging of these the three viewpoints of information, functions and agents leads us to the object-relationship model presented in the next section.

4.3 Object Centred

A natural extension of all of the above models can be captured using a model based on object-oriented design. This is shown in figure 2.

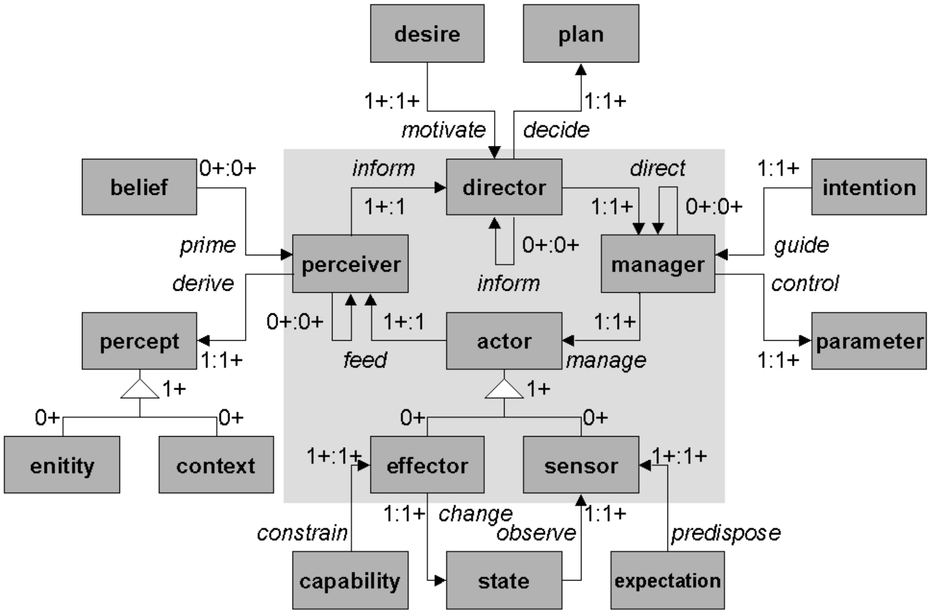


Figure 2: The first two perspectives in the object-relationship model of data fusion.

This object-relationship diagram shows the agent and information perspectives of the model. The notation is similar to that used in object-oriented design and is as follows:

- a class is shown in boldface in a grey box – an *actor* is an example of a class
- A class may be composed of several component classes connected to it via a triangle – for instance a *sensor* is a component class of an *actor*
- Component classes have cardinality constraints – for instance an *actor* may possess no sensors or many sensors
- A class may have cardinality constraints – for instance an *actor* must have at least one of *effector* or *sensor*
- Two classes may be directionally related by a function – for instance *inform* is a function of *perceiver*
- A function has cardinality constraints – for instance each *director* may be informed by one or more *perceivers* and zero or more other *directors*

At first the model in figure 2 may appear complex, but can be viewed from of a set of concentric perspectives. This is its simplicity and also its power since it can be gradually refined all the way to a software code and hence to a deployed system. At the core a fusion system is described as a set of *agents* (computers or people) that are arranged in the same sequencing loop as the earlier functional models. From the information perspective fusion is described as a set of data abstractions which map directly onto the JDL and other models. The level of abstraction is a consequence of the agents, their ordering and their functions, rather than being imposed as an intrinsic property of the system. By allowing the *state* class to contain information about this information as well as the environment the loops-within-loops alluded to by Bedworth[4] are easily included. This captures well the ideas that *"one man's knowledge is another man's data"*[9]. Although figure 2 shows only two layers, this layering of perspectives can continue until the system is deployed (and then further to assist in its maintenance and replacement). See table 1 and figure 3.

Perspective layer	What is described	Example
Agent	Processors and sequencing	Perceiver informs director
Information	Abstraction levels	States, entities and contexts
Theory	Theories and knowledge	Characteristics of a visual sensor
Instance	Methods and formats	An image format
Realisation	Algorithms and data	A particular image

Table 1: Examples of the perspective layers in the object-relationship fusion model.

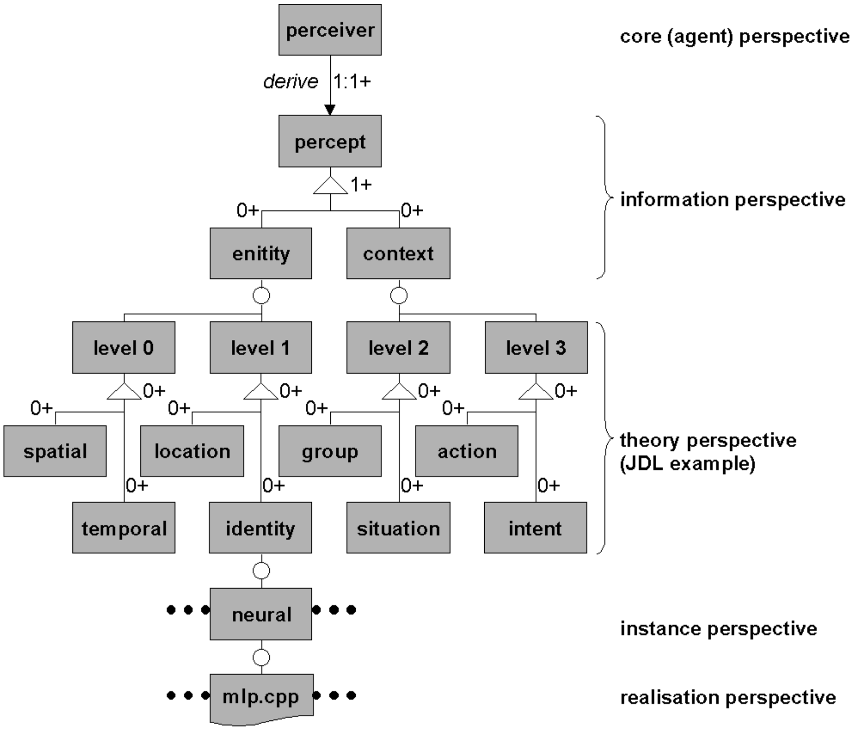


Figure 3: An example of refinement of the object-relationship fusion model, in this case following the JDL philosophy.

This way of thinking is powerful because it may be reasoned about from the core out, that is at all stages in design, development and implementation.

5. RELATIONSHIP ANALYSIS

It is important to separate the system knowledge (the data and procedures which operate on that data within the system) from the state of the general knowledge (the availability and quality of data or the specifications of the procedures). The system knowledge will only become available once a version of the system is constructed whereas the state of the general knowledge will be available during the initial design process. This general knowledge will be input to the development process, while the system knowledge will be the output of this process. Consequently, in the development process the knowledge must be manipulated.

As we stated earlier in this paper, this feature is a distinguishing one for our approach to fusion: we consider fusion as an automatic (or at least semi-automatic) manipulation of knowledge structures, rather than just data. This point of view raises the level of abstraction – instead of considering algorithms that manipulate data objects, we need algorithms that manipulate algorithms (which in turn manipulate data). As a consequence of this departure point, we need methods that can provide an ability to reason about the performance criteria of fusion systems that are being designed. This kind of ability is typically achieved through the analysis process carried out by the (human) developer of the fusion system (during design) and through simulations (during system verification). The requirement on "reasoning" in the context of the human makes an implicit assumption that the human is responsible for managing the degree of rigor in the reasoning process. The human is expected to carry out a sound reasoning process, rather than just guessing. The human is also expected to relax the rigor, to an acceptable level, whenever there is a lack of information that would be needed for a fully sound process. This kind of a requirement is very difficult to implement in a mechanical reasoner. Typical approaches to this problem are on the edges, either a fully consistent process like logic (but then we encounter the known problems of undecidablity) or a heuristic approach like expert systems (but then we cannot guarantee consistency). In order to deal with this problem, we selected a formal method approach based on category theory.

Within the framework of category theory we can address all of the problems stated above. In particular, we can analyse the effects of combining (fusing) structures (*e.g.*. algorithms) into one structure with respect to the system level criteria. In other words, we can analyse various design choices. The input to this fusion process is the state of the general knowledge (as stated above) and the system level requirements (we also refer to this as knowledge structures). The development consists of a number of steps of structure fusion. Since each of the steps is provably correct, *i.e.* the resulting structure is guaranteed to preserve the system requirements of the previous step, the whole process is provably correct. The process consists of two phases: specification development and refinement. The specification development phase ends with a complete specification of the system. The refinement phase takes such a complete specification and ends with code. Again, since only formal, provably correct operators are used in the refinement process, the resulting code is provably correct with respect to the specification. In the rest of this section, we give a glimpse of our approach. More information on this subject can be found in our paper. [12]

A *category*[24]is an abstract mathematical construct consisting of *category objects* and *category arrows*. Category objects are the objects in the category of interest. In our case they are algebraic specifications of computational objects; the category is called *Spec*. Category arrows define a mapping from the internal structure of one category object to another and are also called *morphisms*. In our case they are *specification morphisms*. In the category *Spec*, specification morphisms map the sorts and operations of one algebraic specification into the sorts and operations of a second algebraic specification such that the axioms in the first specification become provable theorems in the second specification. Thus, in essence, a specification morphism defines an embedding of one specification into a second specification.

Specification morphisms are required for defining and refining specifications. Additionally, we need the *combination*, or composition, of existing specifications to create new specifications. This is where category theory is extremely useful in information fusion. Often two specifications that were originally extensions from the same ancestor need to be combined. Therefore, the desired combined specification consists of the unique parts of two specifications and some “shared part” that is common to both specifications (the part defined in the shared ancestor specification). This combining operation is called a *colimit*[24]. The colimit operation creates a new specification from a set of existing specifications. This new specification has all the sorts and operations of the original set of specifications without duplicating the “shared” sorts and operators.

Diagrams are the main modeling tool used to represent category objects and category arrows. A diagram consists of nodes representing category objects, and arrows, representing category arrows. In our approach, the category objects are the specifications (structures) of sensors, objects of interest (*e.g.* targets), goals of the system, general knowledge, and system knowledge. A whole system is represented as one diagram. Once such a diagram is developed, one can reason about the properties of the system by using *theorem proving*. In other words, one needs to state a hypothesis and then prove or disprove the hypothesis within the knowledge included in the diagram.

An example of a diagram for a simple object recognition system is presented in Figure 4. In this figure, the top category object is termed WORLD. It contains specifications of the world constraints, *e.g.,* that two (world) objects cannot appear in the same place at the same time and what are the expected types of objects in this world. This specification embeds the specification called RECT-TRIAN-SHADOW, represented as a morphism between the two category objects. RECT-TRIAN-SHADOW, in turn is formed as a colimit of three specifications below. The lower layer represents specificaitons of the two sensors involved – RANGE-SENSOR and INTENSITY-SENSOR. At the bottom there are specifications of mathematical objects needed to model the sensors.

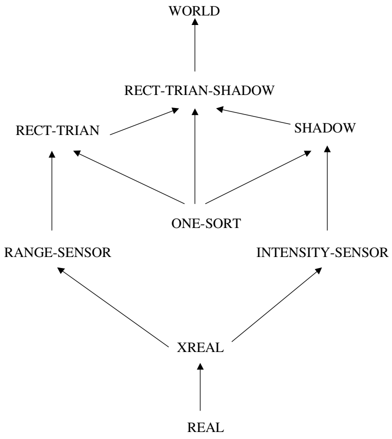


Figure 4: A diagram of an information fusion system

As we have shown in our earlier work[11], formal methods (such as category theory) provide a reasoning framework that allows to analyse how abstract, system level performance criteria can affect design choices. Formal methods have recently been successfully applied to the analysis of relatively simple data fusion systems. In order to apply formal methods to design systems for real-word applications of typical complexity, system solutions and theories are needed that constrain the design space. In this paper we outline an approach to combining these methods and tools with other methods and tools that are relevant to the development of information fusion systems.

6. INFORMATION DYNAMICS

Petri-net modelling is an abstract, formal way of modelling information flow in concurrent systems[13], which were proposed as a means of representing asynchronous parallel processing. A Petri-net is a directed bipartite graph of information processors (which are called *transitions* and drawn as bars in Petri-net diagrams), the information stores (called *places* and represented by circles) and the information itself (in the form of *tokens*). In coloured Petri-nets the tokens are typed (*coloured*) and the information dynamics that can be modelled are somewhat richer[14]. In both the standard and coloured versions, a substantial arsenal of mathematical tools has been built up to enable the detection of waiting, blocking and deadlock. There are a number of commercial and freeware Petri-net tools which can simulate systems and perform these kinds of analyses[15].

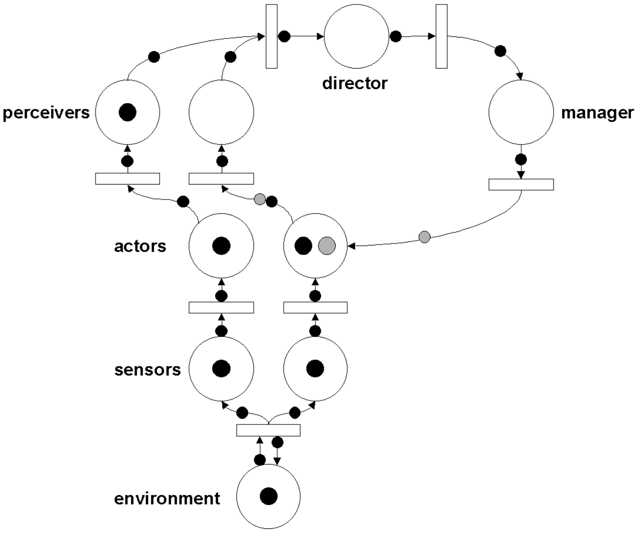


Figure 5: A marked, coloured Petri-net representation of a fusion system.

In figure 4 a coloured Petri-net representation of a particular fusion system is shown. In this case the environment supplies data to the two sensors simultaneously. The sensors are on separate platforms (actors), one of which is autonomous and the other is under system control. Separate perceptions are formed from the two information sources and the fusion takes place at this perception stage (which may be feature-level fusion or soft decision-level fusion). The decisions made by the director are implemented by a just one manager. Using appropriate simulation and analysis tools it is possible to determine that this system will not deadlock and that the critical loop is the control loop on the second actor.

7. HUMAN ASPECTS: PLUG-AND-PLAY *WELL* TOGETHER

The power of an object-oriented architecture lies, in part, in the ability to populate any of its objects with any instance that satisfies the specifications for the given object. The proposed object-oriented architecture is one in which the specifications have historically been job descriptions for actors, perceivers, directors and managers, with the instance objects being human beings who attempt to satisfy their job descriptions. Only in the past couple of decades has there been a serious attempt to instantiate this architecture's component objects with automated systems that attempt to satisfy some substantial part of the job descriptions. As fusion architectures' objects are populated with more complete automations working alongside people, each automated object becomes progressively less of a mere tool and more of a co-worker.

Social Competence and Inference. Whether in a humans-only system or a mixed human-automated system, the people are typically mature adults who are already fully trained to exhibit adequate adaptive competencies (including social competencies). People normally bring to the workplace an entire repertoire of responses for social interaction. These are usually highly differentiated, highly articulated and highly automated. Properly exploited, peoples' established repertoires offer a substantial advantage. With little (if any) additional training, people can immediately apprehend the intentions of other autonomous agents within their shared context and position themselves appropriately. To leverage people's existing social competence, each automated object similarly needs to position itself correctly, cueing its human counterparts within social norms, and responding to peoples' cues within the same norms. To the extent that an automated object satisfies normal social specifications, it is likely to fit into a fusion architecture with its human counterparts as seamlessly as any human could be said to fit in.

There, however, is the rub. Game theory would seem to suggest that, as specifications for behaviour, job descriptions would serve as a substantial body of shared information that would insure well-oiled co-operative social relationships in which every person knows how to fit themselves in. Yet, as a result of the broad variability of peoples' aptitudes and intentions, people (no less machines) rarely fit seamlessly into complex social organisations. It is long settled psychology that a substantial part of the human social repertoire is focused on apprehension of the nature of one's own and others' fit into the social context[16].

Identifying the redundancies of others' intentions with one's own is the foundation for trust. With respect to their own perceived interests, people want to know if their fellow social agents are co-operative or competitive, reliable and trustworthy or dangerously unpredictable. In addition to the first order problem of understanding how to behave to satisfy the specification of some job description, people put substantial energy into monitoring the performance of others. By engaging in a second order analysis of the competencies and intentions of others, people try to determine if others making an adequate fit and, more importantly, whether they are trying to. Because humans are a social species the adaptive competence of whom is closely tied to competent social functioning, the second-order analysis of others' aptitudes, reliability and intentions is so fundamental to human adaptive competence that it is highly automated and often largely unconscious. Peoples' automated responses to each other are first and foremost emotional responses, with rationales lagging behind (if they follow at all).

Emotional Governance. The emotionality that automates social behaviour results from emotions' role as control signals in peoples' adaptively competent information processing[17,18]. Figure 6 (originally in Frankel[17]) shows an adaptive (feedforward) control topology that is closely related to the topology of the fusion architecture proposed in this paper, with two exceptions:

1. adaptive controls are explicitly articulated
2. data abstraction is treated as an orthogonal problem to control.

In Figure 6, ***emotions*** are shown to control the feedforward process of adapting to changing circumstances. *Salience:* Emotions control salience, appraising the urgency of changing situational contingencies and interrupting current activities to draw attention to events of equal or greater priority. *Categorisation:* Emotions control evaluative categorisation, appraising event type and vectoring memory access to a palette of relevant response options from the individual's repertoire. *Response Selection:* Emotions control response selection, appraising both current harm and benefit and also the expected harm and benefit associated with the response options in the palette. Emotions then drive the selection of what is expected to be the most favourable (or least unfavourable) response. *Settling:* Emotions control the rate of settling on selected responses. Emotional confidence appraises (*a*) that the selected response is a good fit, (*b*) that the response is commensurate with available abilities, and (*c*) that the cost of an inappropriate response is low. Conversely, anxiety appraises (*a*) that the selected response is still a poor or uncertain fit, (*b*) that situational demands exceed abilities, or (*c*) that the cost of an inappropriate response is high. Error detection is dampened in inverse proportion to anxiety (or in direct proportion to confidence, where a region of certain knowledge disables all relevant error checking).

Emotions embody, automate and signal to others the appraisals and intentions that organise and motivate behaviour. In effect, emotions are the control and social signals in the adaptively competent information processing architecture that governs real-time human fusion.

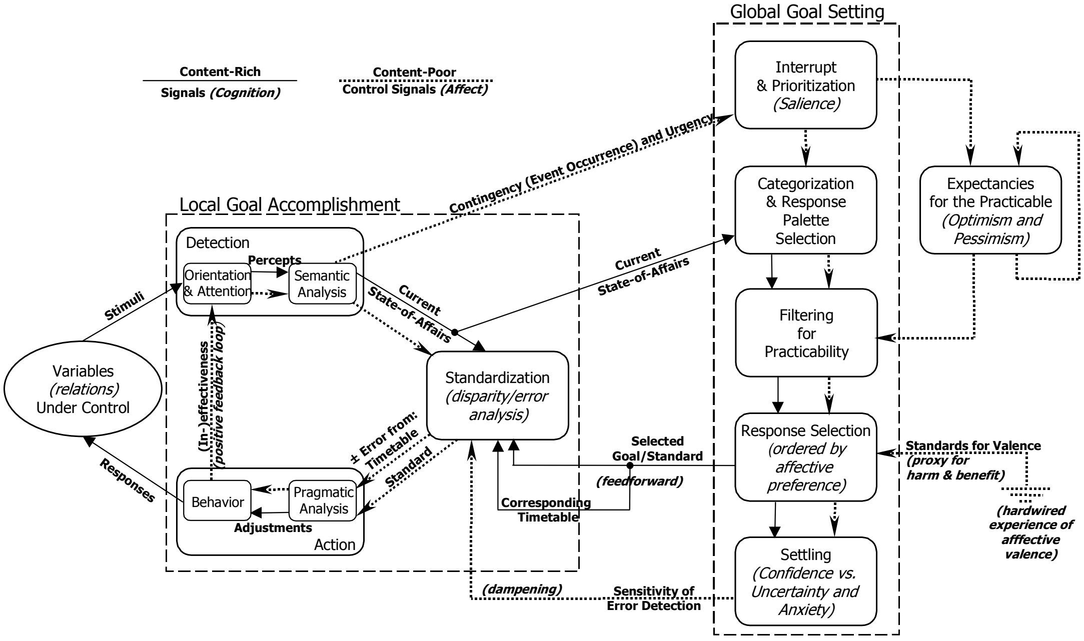


Figure 6. Affectively Governed, Expectancy Biased Adaptive (Feedforward) Control: An Architecture for the Adaptively Competent Fusion in the Human Individual

If emotions govern human fusion internally and signal human fusion externally, then automated fusion objects can, in principle, fit into fusion architectures as seamlessly as people do. However, automations will fit in only to the extent that they *both* satisfy their job descriptions *and also* participate appropriately in the exchanges of emotional display by means of which people read each others' intentions. One implication, not treated here, is that automated co-workers should both cue human co-workers with emotional displays and also utilise people's emotions, thereby constructing a shared social narrative between automation and person. The other more tractable implication is that, beyond satisfying the job description, the task-oriented behaviour of automated systems must satisfy peoples' social expectations, so that people will come to trust the automation.

Much study is needed in this area; however, existing psychology points to emotional factors that can undermine trust and render fusion automations under-utilised. A few such factors follow.

Settling. In any feedforward control problem with significant performance pressure, there is always a tension between selecting the best reference for changed contingencies and settling on some adequate reference in a timely fashion. For people, the same anxiety that slows settling due to low confidence also punishes slow settling, decreasing the amount of information taken in and processed[19], and eventually increasing impulsive settling[19]. Because people apprehend the variability of human behaviour, they become confident of others based on experience with others *in the range of circumstances* likely to be encountered together with others. In this way, people come to learn when to depend upon others' aptitudes and when to avoid their ineptitude, so that it is possible at all events to settle quickly. Automated fusion objects should thus support simulation over their entire operating range, so that, through training, people can come to trust their own judgements of automation performance and maintain a settled relationship with automations' behaviour. Moreover, training regimens must trade-off the need to focus on the most commonly occurring scenarios with the likelihood that trust of the automation may break down more quickly, exactly when the automation is most needed, that is, under the anxiety of unfamiliar situations in which the automation is exhibiting unfamiliar or unexpected (but correct) productions.

Editing. It is the function of emotions to make unusual and anomalous events salient, so as to generate considered responses. However, a side-effect of salience of unusual events is an editing effect[21] that makes unusual events seem more likely, a distortion of the probability distribution against which an automation is likely to be robust. Human and automated actors can become significantly mismatched in the orientation of their vigilance. Moreover, in the course of transformation from the concrete domain of actors to the abstract domain of directors, small divergences in the content of percept streams can become amplified into large divergences of situational analysis. Thus, the judgements of a less distortion prone automation can come to diverge significantly from those of a more distortion prone human counterpart, potentially reducing trust.

Regret-avoidance. People are regret averse[22], and can fall into protracted "sunk-cost" behaviour[22], increasing risk-taking in the course of trying to make up for losses incurred. Both director and manager automations that are more robustly rational in cost-benefit analyses and decision-making can make recommendations that people may experience as depriving them of the opportunity to cover their losses. This may cause the panic, anger and distrust that people often feel when they perceive others to be thwarting their attempts to avoid feeling (and looking) bad.

7. SUMMARY

We have outlined a relationship model for data fusion based on object-oriented principles that goes beyond existing *descriptive* models and provides a *prescriptive* model that can be gradually refined to the point where it can be deployed. The model is amenable to analysis at the design stage through the use of formal methods. The information flow in a design may be studied using Petri-net theory. The human aspects of information fusion systems are easily accommodated within the model, since no distinction is drawn between human and computer objects until model is viewed from the realisation perspective.

The orthogonality of abstraction and control has not yet been fully integrated into our model. This needs to be resolved before moving on. The design approach also needs to evolve from a largely theoretical model to a practical prescription for efficient system design. The associated tools need to be honed to provide tractable analyses of realistic, full-scale problems. As these analysis tools develop in a fusion setting a network of specifications and theories will need to be established and the fusion object model populated with specifications for generic fusion objects (at least) from the agent and information perspectives. Particular specialisations can then inherit the attributes and behavioral characteristics of these objects. The practical experience acquired using a case study has much to offer and we should look towards applying this design philosophy to a medium-scale system in the near term.

Recently, the Unified Modeling Language (UML) has become a *de facto* standard for representing software systems. The object diagram shown in this paper does not use the UML notation. It may be useful, for the sake of reaching a wider software engineering community, to translate this diagram into UML. Also, we may incorporate some aspects of the Unified Software Development Process (USDP) to capture the evolution of a fusion system – from specification to implementation. Addtionally, in order to combine such a diversity of representations of a fusion system as described in this paper, we may employ the concepts of various *views*, as promoted by the Reference Model for Open Distributed Processing (RM-ODP), in which five different views of a system are used to represent the interest of various stakeholders.

Finally, the underpinning psychological theories for dynamic teams of mixed human and computer co-workers is far from established. The problem is not so much that mixed human-computer teams are poorly understood. What is known about human-computer interaction suggests that people fairly automatically parse the computer the same as any other autonomous agent, projecting a personality onto the computer and behaving as they would with any other social agent. The computer, however, is only very rarely programmed with the sophistication to be a credible social agent, and so eventually comes to take on the personality of a flaky, frustrating agent that is warily trusted more as a matter of necessity than of choice. The problem, rather, is that psychological theory regarding purely human teams is still in its infancy and is quite far from being any kind of settled science. As such, the challenge is to understand the social expectations of the humans, so that the computer can live up to them, and so that, where the computer can outperform the humans, the computer can successfully elicit peoples' genuine, settled trust. Further research in this area will be necessary before designs based on the object model can properly integrate humans and machines as peers.

ACKNOWLEDGMENTS

The authors wish to express their gratitude to Jane O'Brien of the University of Central England for her contributions to this paper. The first author wishes to acknowledge support from AFOSR/NM, Grant No: F49620-98-1-0043.

REFERENCES

1. D. L. Hall, "Multisensor Data Fusion and the Millenium: Are You Y2K Compliant?", supplement to the Proceedings of 2nd European Conference on Information Fusion, EuroFusion99 (1999).

2. D. L. Hall and A. K. Garga, "Pitfalls in Data Fusion (and How to Avoid Them)", Proceedings of 2nd International Conference on Information Fusion, FUSION'99, Sunnyvale, USA, pp. 429-436 (1999).

3. A. N. Steinberg, C. L. Bowman and F. E. White, "Revisions to the JDL Data Fusion Model", Proceedings of SPIE AeroSense (Sensor Fusion: Architectures, Algorithms and Applications III), Orlando, Florida, USA, pp. 430-441 (1999).

4. M. D. Bedworth and J. C. O'Brien, "The Omnibus Model: A New Model for Data Fusion?", Proceedings of 2nd International Conference on Information Fusion, Sunnyvale, USA, pp. 437-444 (1999).

5. B. Dasarathy, "Sensor Fusion Potential Exploitation – Innovative Architectures and Illustrative Applications", Proceedings of IEEE, Volume 85, Number 1, pp. 24-38 (1997).

6. M. Markin, C. Harris, M. Bernhardt, J. Austin, M. Bedworth, P. Greenway, R. Johnston, A. Little and D. Lowe, "Technology Foresight on Data Fusion and Data Processing", Publication of The Royal Aeronautical Society, (1997).

7. A. Shulsky, "Silent Warfare: Understanding the World of Intelligence", Brassey's (1993).

8. J. Boyd, "A Discourse on Winning and Losing", Maxwell AFB Lecture (1987).

9. F. E. White, personal communication (1998).

10. H. Gao, M. M. Kokar and J. Weyman, "An Approach to Automation of Fusion using Specware", Proceedings of 2nd International Conference on Information Fusion, Sunnyvale, USA, pp. 109-116 (1999).

11. J. Li, M. M. Kokar and J. Weyman, "Incorporating Uncertainty into the Formal Development of the Fusion Operator", Proceedings of 2nd International Conference on Information Fusion, Sunnyvale, USA, pp. 125-132 (1999).

12. M. M. Kokar, J. Tomasik and J. Weyman, "A Formal Approach to Information Fusion", Proceedings of 2nd International Conference on Information Fusion, Sunnyvale, USA, pp. 133-140 (1999).

13. J. L. Peterson, "Petri Net Theory and the Modelling of Systems", Prentice-Hall (1981).

14. J. Ferber, “Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence”, Addison-Wesley (1999).

15. University of Aarhus Department of Computer Science, “The World of Petri Nets“, http:/www.daimi.au.dk/PetriNets (2000).

16. M. D. Maselli and J. Altrocchi, “Attribution of intent”, *Psychological Bulletin, 71,* 445-454. (1969).

17. C. B. Frankel, “Such Order From Confusion Sprung: Adaptive Competence and Affect Regulation”, Dissertation presented to the faculty of Pacific Graduate School of Psychology, Palo Alto, CA (1999).

18. C. B. Frankel and R. D. Ray, “Affect and the Architecture of Adaptively Competent Information Processing”, Paper accepted for and to appear in the proceedings of the 2000 conference of the Society for the Study of Artificial Intelligence and the Simulation of Behaviour, Birmingham, England, two-day symposium entitled “How to Design a Functioning Mind”, (2000).

19. G. Keinan, N. Friedland, and L. Arad, “Chunking and integration: Effects of stress on the structuring of information”, *Cognition and Emotion, 5,* 133-145. (1991).

20. G. Keinan, “Decision making under stress: Scanning of alternatives under controllable and uncontrollable threats”, *Journal of Personality and Social Psychology, 52,* 639-644. (1987)

21. .D. Kahneman, and A. Tversky, “Choice, values and frames”, *American Psychologist, 39,* 341-350. (1984)

22. .R. P. Larrick, “Motivational factors in decision theories: The role of self-protection”, *Psychological Bulletin, 113,* 440-450. (1993).

23. M. Zeelenberg and J. Beattie, “Consequences of regret aversion 2: Additional evidence for effects of feedback on decision making”, *Organizational Behavior and Human Decision Processes, 72,* 63-78. (1997).

24. S A. DeLoach and M. M. Kokar, “Category Theory Approach to Fusion of Wavelet-Based Features”, Proceedings of 2nd International Conference on Information Fusion, Sunnyvale, USA, pp. 117-124, (1999).