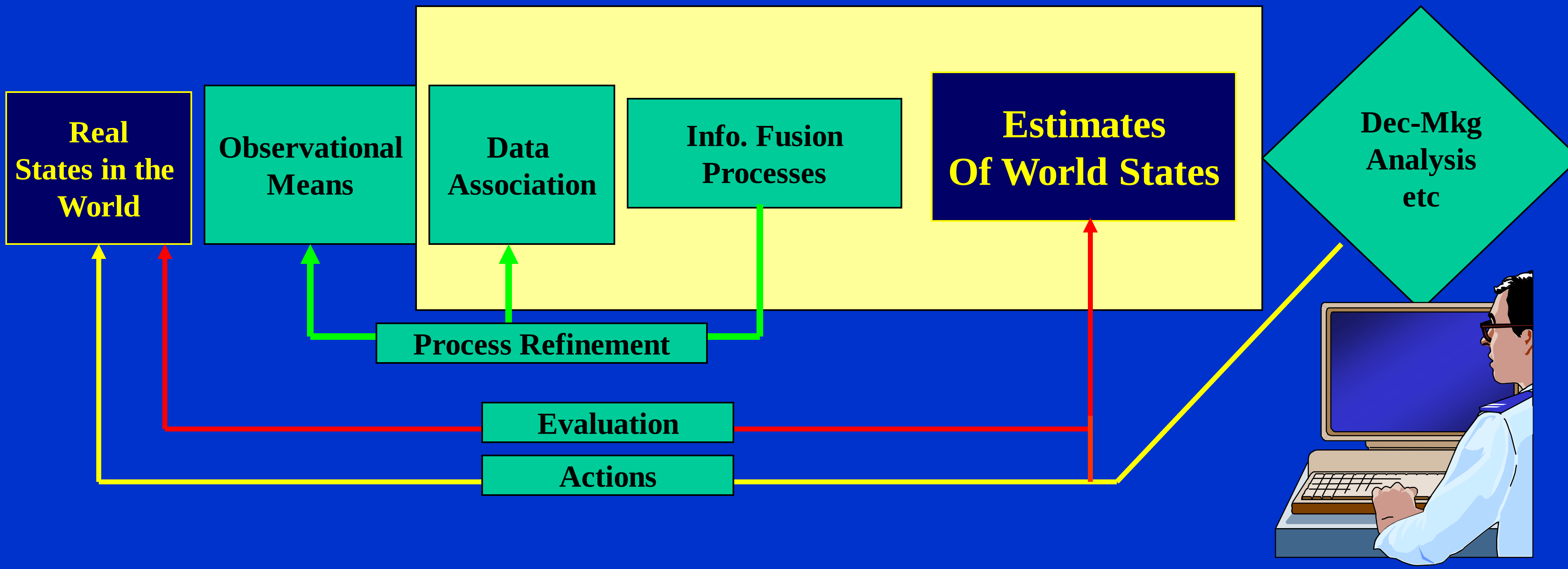
A Survey and Analysis of Frameworks and Framework Issues for Information Fusion Applications

James Llinas
Santander Chair of Excellence
Universidad Carlos III de Madrid
HAIS'10—5th Intl Conf on Hybrid AI Systems, San Sebastian, Spain
June 25, 2010

center for multisource informationfusion
A CUBRC and University at Buffalo Research Center

Universidad Carlos III de Madrid

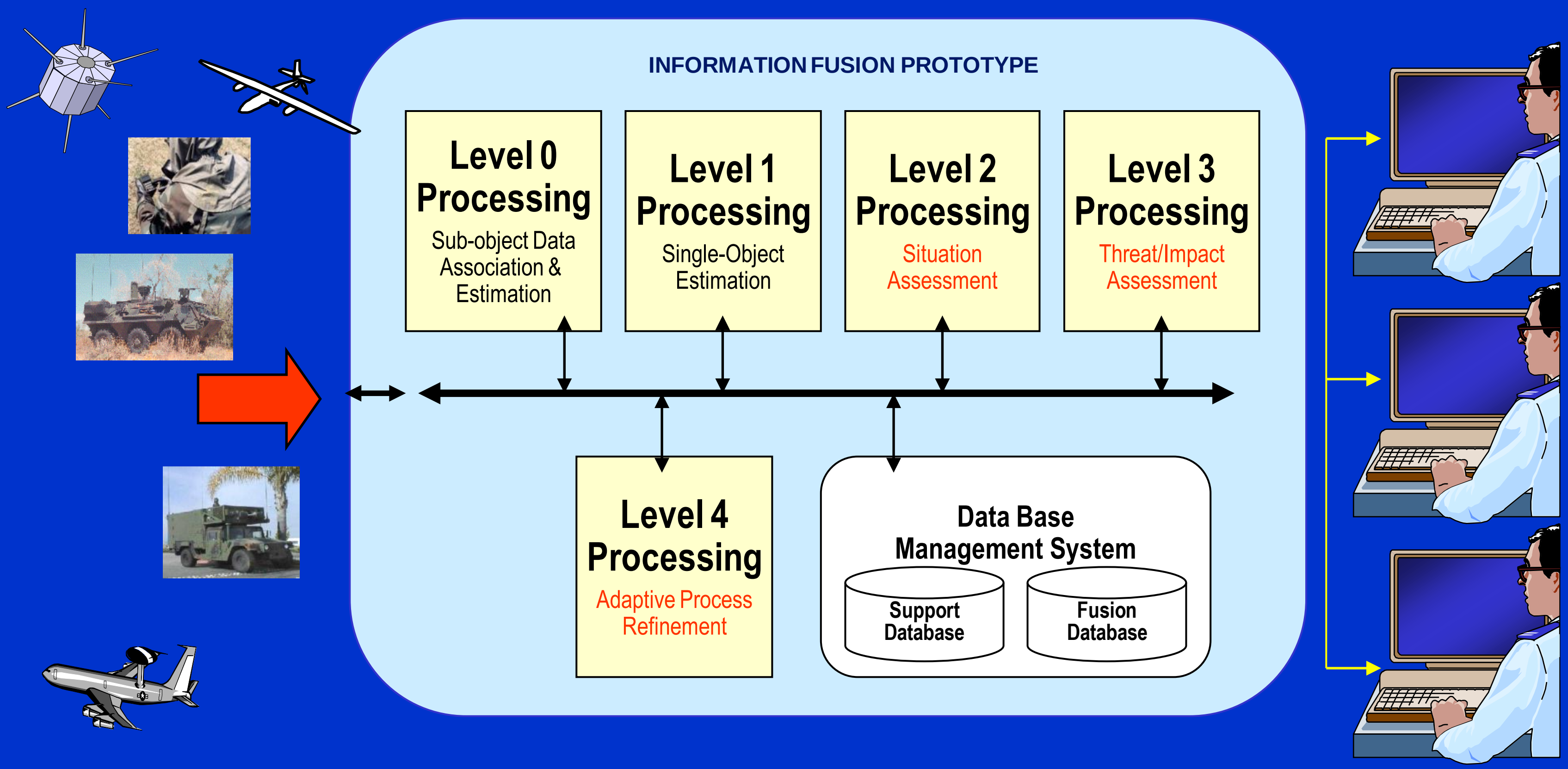
Nature of Information Fusion



- One means to satisfy user information needs for decision/analysis support, i.e., most frequently inserted to support human user

Data Fusion Functional Model

(Jt. Directors of Laboratories (JDL), 1993)



Framework Thoughts to Date

- Centralized Architecture
- Framework is Fusion-Level-based, not Sensor-based
  - Assumes Sensor Data and Sensor-based Estimation-to Fusion Level partitioning defined
    - Defines Data and Estimation Flow to Fusion Nodes and Levels
    - If feasible, allows for soft-switching of this flow control
- Each Level responsible for Within-Level Optimization ("Greedy" approach at each Level)
- No learning, knowledge mgmt, adaptive model mgmt
  - Deductively-based, assumes A Priori Dynamic World Model exists
- No User Interface
- No Humans in the Loop
- Architected Baseline with Defined Inter-Level Dependencies and Contingency-sets
  - Influences adaptive logic

Nature of a Framework

- Definitions (General Framework)
  - "A structure for supporting or enclosing something else"
  - "conceptual structure intended to serve as a support or guide for the building of something that expands the structure into something useful"
  - Thus, not intended, by design, to be "useful" but to help build something useful
- Software Framework
  - "an abstraction in which common code providing generic functionality can be selectively overridden or specialized by user code providing specific functionality".
- Should not have domain-specific components
- **"inversion of control" –Framework imputes (main) control structure for any application**
  - Important aspect: Framework must allow all desired controllability and adaptability

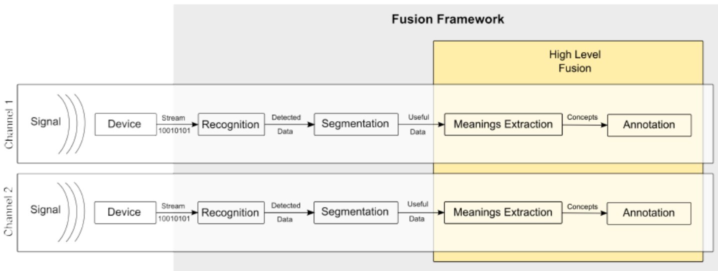
Literature Review on Fusion Frameworks

Paper [Refs]	Framework Focus	Cited advantages	Disadvantages/issues
Besada	Real-time apps	--Oriented-graph architecture --GUI-based algorithm selection	--Currently limited to Level 1 type functions --Does not discuss reqmt to have algorithmic performance profiles
Bolles	Intelligent auto apps	--Employs data stream mgmt techniques	--Automotive-application specific
Dastner	Development suite vs a framework	--Object oriented --Plug-in modules --Uses Fusion Node	--Focused only on Level 1 --Details not shown
Emami	Significant human involvement in a toolkit concept	No specific IF substructure	No specific IF substructure
Hou	Target recognition apps	See separate discussion on Blackboards	See separate discussion on Blackboards
Julier	Networked/distributed Sensor/Fusion nodes	--Agent approach; use COABS grid approach	--Mostly Level 1 oriented, only numerical operations
Klausner	Embedded-system apps	--Presumes powerful individual sensor nodes --Somewhat BB-like --All fusion Levels	--Fusion abstracted as holistic process; no substructure --No within-node framework
Kokar	IF system and process specification	No process framework offered	No process framework offered
Kumar	Wireless ad hoc sensor networks	--Automatically managed placement of fusion services --Fusion API for fusion fcts and data flow --Fusion as directed task graph	--No consideration of DDF issues such as OOSM and incest --Optimization is largely directed to network factors balanced against fusion performance

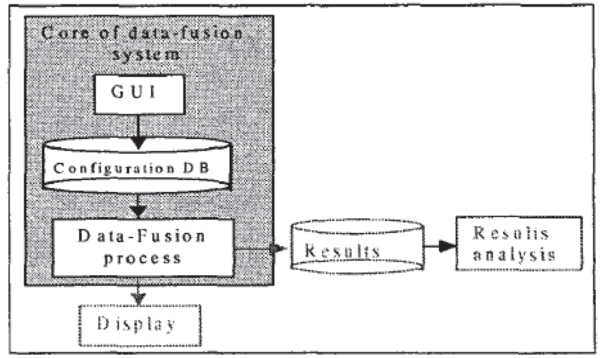
Literature Review on Fusion Frameworks

McDaniel	Use of IF for integrating disparate DB data sets	No process framework offered	No process framework offered
Mendoca	Framework abstraction for ambient intell type apps	--Only framework paper that addresses multi-modal inputs,	--Restricted range of application domains --
Mirza	Overview of several major fusion architectures	No process framework offered	No process framework offered
Paradis, Roy	Robust, fusion-based simulation environment	--Blackboard selected; usual BB features cited	--Mainly focused on Naval apps
Posse	Mathematical characterization of humans interacting with fusion processes	No process framework offered	No process framework offered
Rothenhaus	Software pattern characterization	-- SOAextensibility. --Trickle-up software design pattern to decouple data management from fusion -- Zone pattern provides a view of the relationship and roles between functions	For both patterns: --Performance impacts -- Requires common data schemas and definitions to support late binding and orchestration data mgmt and fusion operations --Imputes software overhead --Complexity --Configuration mgmt
Sycara	High-level fusion for Army-type, force-on-force military engagements	--Incorporates contextual aspects --Focused on high-level fusion --Multi-agent approach --Incorporates IPB methodology	--Militarily-specific (totally committed to IPB method flow) --Really a robust point design for force-on-force high-level fusion

Process/Function Structures from the Literature



Mendoca



Besada

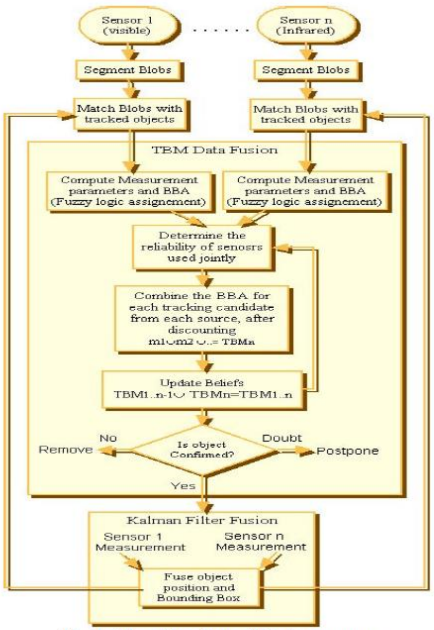
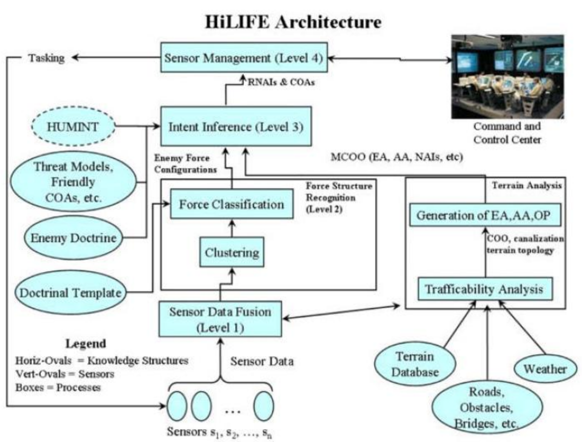
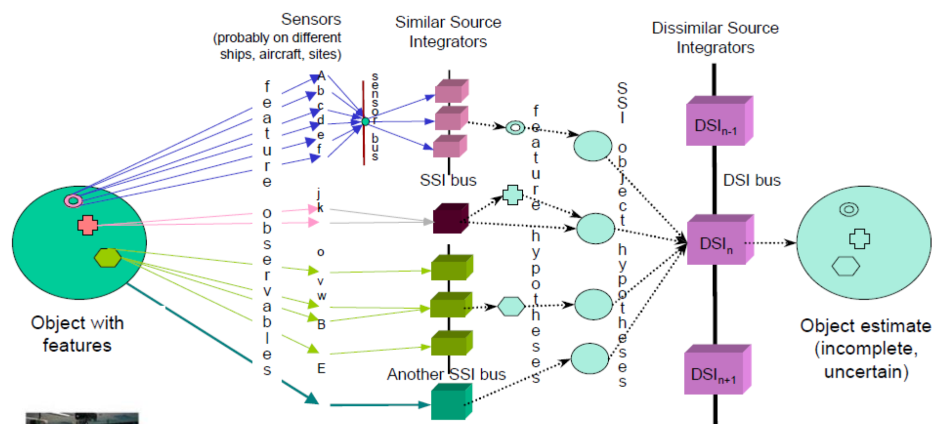


Fig.1. Algorithm flow of Fusion process

Kumar



Sycara



McDaniel

Literature Review Summary

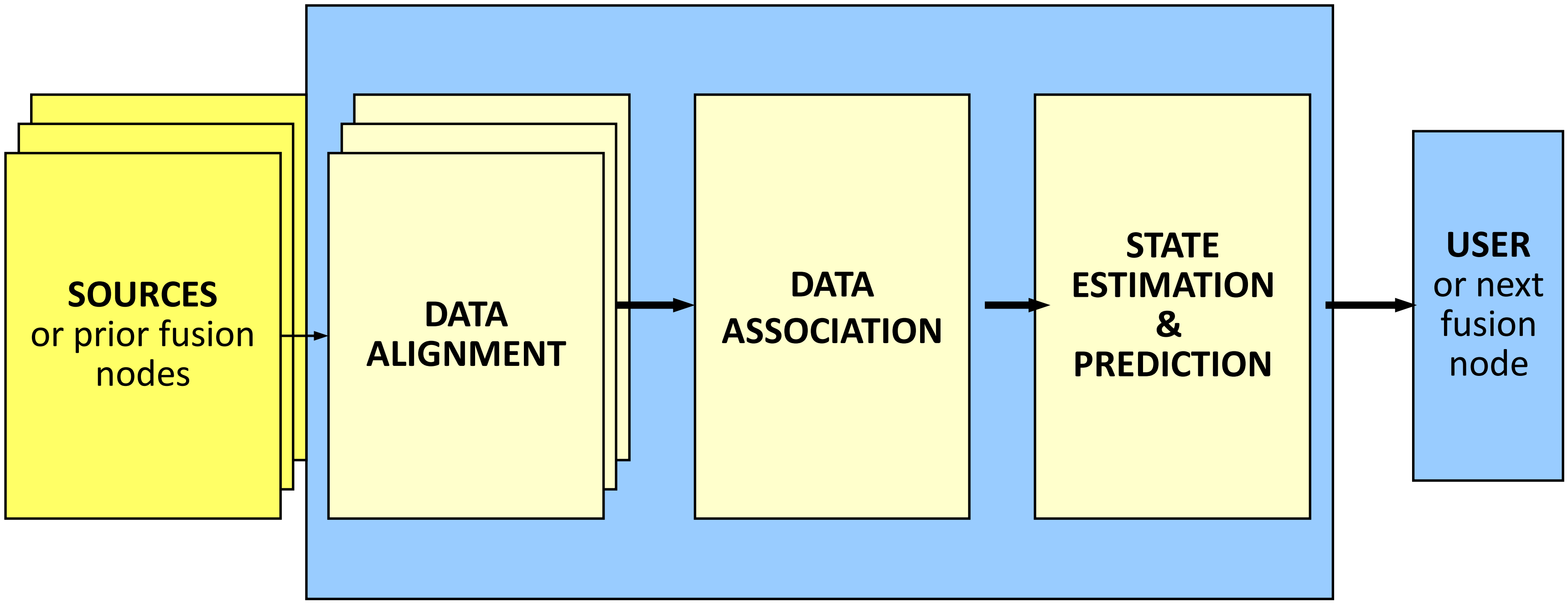
- Widely-varying levels of abstraction
- Many do not address specific notion of a processing Framework definition
  - Mix "architecture" with "framework"
- Do not address controllability/adaptability specifically
- Often do not associate to JDL Levels
- Most do not use the "Fusion Node" paradigm

None are Domain-Independent "Build-To" Frameworks

General Strategy

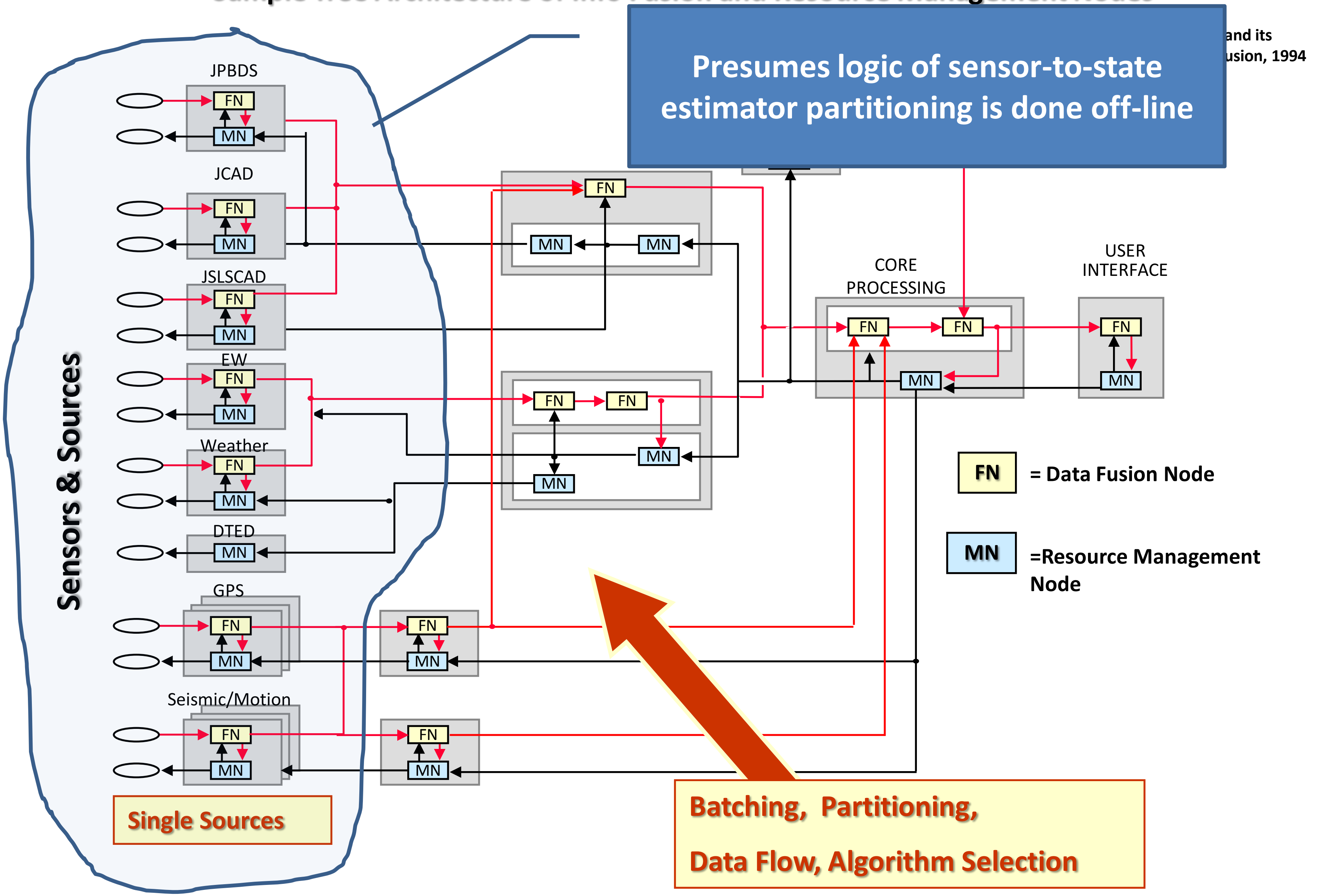
- Any complex Fusion System is a collection of partitioned, fusion-based State Estimators
  - A collection of (possibly-interacting)Fusion Nodes
- Partitioning the overall problem by level of abstraction, ala JDL Levels, remains a reasonable approach
- Ideally implemented, Fusion processes are adaptive in various ways
  - Sensor-input management
  - Data/Estimation flow control
  - Intelligent invocation of multiple algorithms
  - Inter-Fusion Node synergies (eg Tracking ⟷ Classification)
  - Inter-Level Synergies (eg L2 ⟷ L1)
  - Framework must allow Layered Control

FUSION NODE

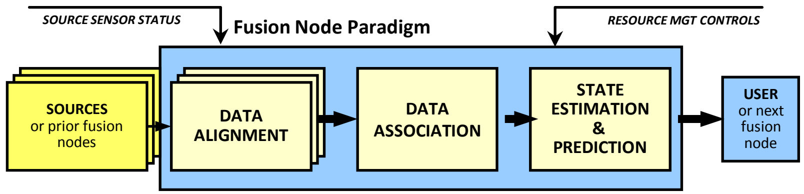

SOURCES
or prior fusion
nodes
DATA
ALIGNMENT
DATA
ASSOCIATION
STATE
ESTIMATION
&
PREDICTION
USER
or next
fusion
node

Architecting Fusion Systems:

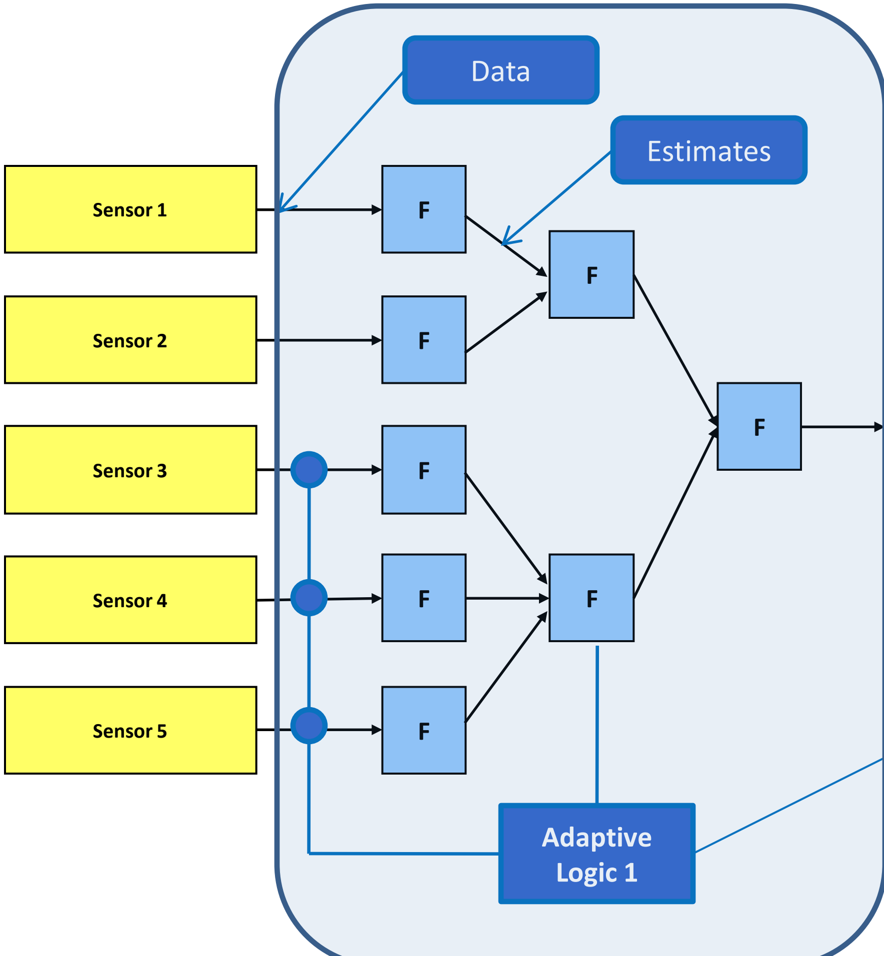
Sample Tree Architecture of Info Fusion and Resource Management Nodes*



Complexities in Fusion Process Architectures



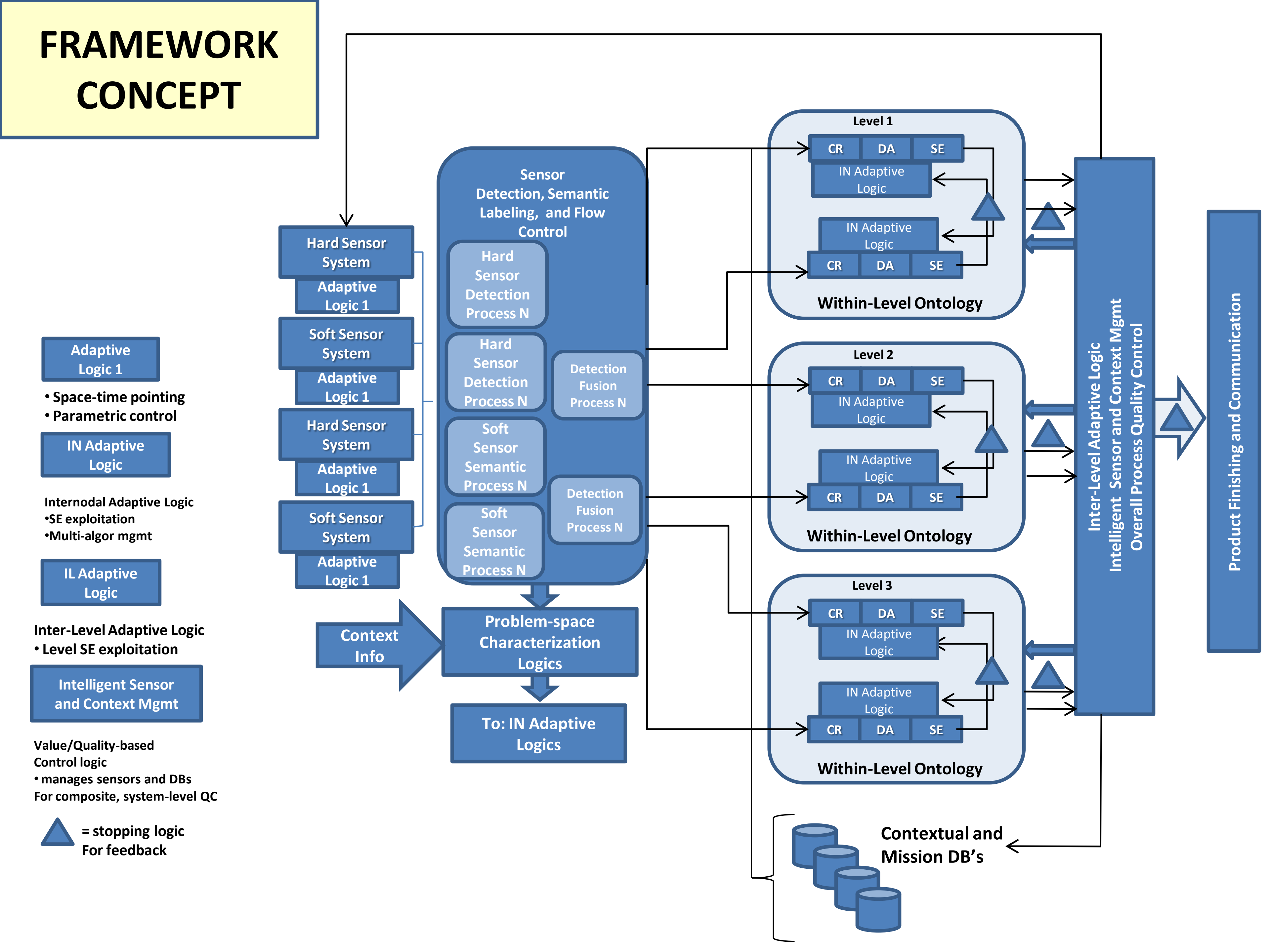
Any given Fusion Node



Framework Development strategy centers on generalizing this structure

Architecting (Batching, Partitioning) of a Real System Involves a Structured Collection of Fusion Nodes

- Centralized
- Locally Distributed
- Hybrid (Data or Estimate Fusion)
- Adaptive -- Various
- Networked (Onboard-Offboard)


FRAMEWORK CONCEPT
Hard Sensor System
Adaptive Logic 1
Soft Sensor System
Adaptive Logic 1
Hard Sensor System
Adaptive Logic 1
Soft Sensor System
Adaptive Logic 1
Sensor Detection, Semantic Labeling, and Flow Control
Hard Sensor Detection Process N
Hard Sensor Detection Process N
Soft Sensor Semantic Process N
Soft Sensor Semantic Process N
Detection Fusion Process N
Detection Fusion Process N
Context Info
Problem-space Characterization Logics
To: IN Adaptive Logics
Level 1
Level 2
Level 3
CR
DA
SE
IN Adaptive Logic
Within-Level Ontology
Inter-Level Adaptive Logic
Intelligent Sensor and Context Mgmt
Overall Process Quality Control
Product Finishing and Communication
Contextual and Mission DB's
Adaptive Logic 1
• Space-time pointing
• Parametric control
IN Adaptive Logic
Internodal Adaptive Logic
•SE exploitation
•Multi-algor mgmt
IL Adaptive Logic
Inter-Level Adaptive Logic
• Level SE exploitation
Intelligent Sensor and Context Mgmt
Value/Quality-based
Control logic
• manages sensors and DBs
For composite, system-level QC
= stopping logic
For feedback

Task-sharing Strategy

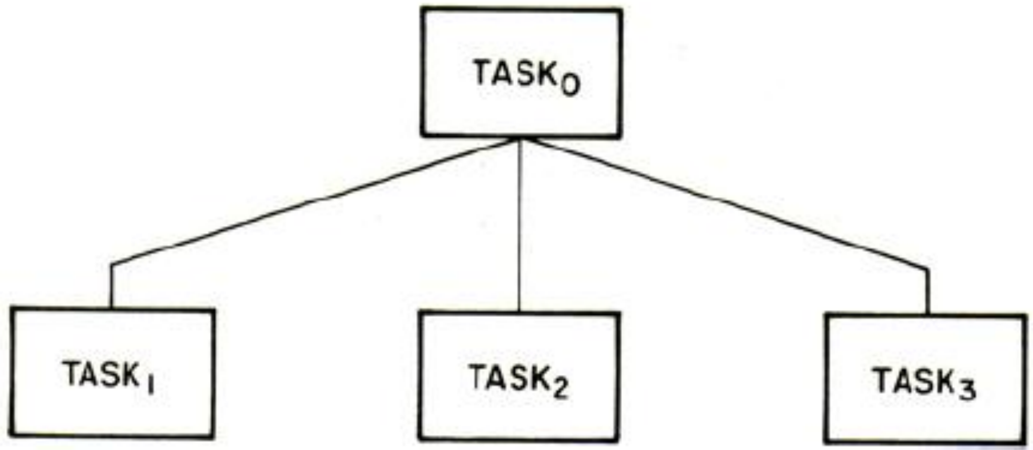


Fig. 2. Task-sharing.

- Control usually Goal-directed
- Key issue: task partitioning, inter-task or process communication
- One communication paradigm: negotiation (eg Contract Net Protocol in FIPA)
- Common Domain KS components in each processor (Common overarching KS)

  --explicit partitioning: limited communication reqmts

Results-sharing Strategy

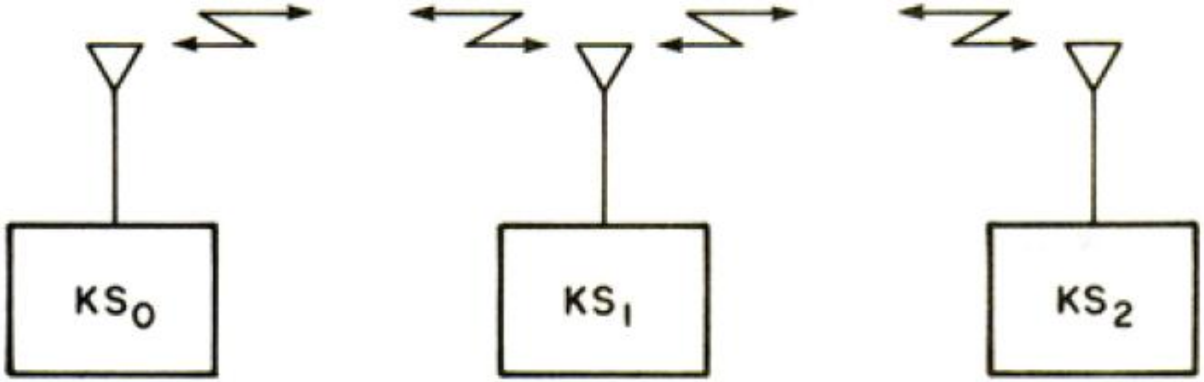


Fig. 8. Result-sharing.

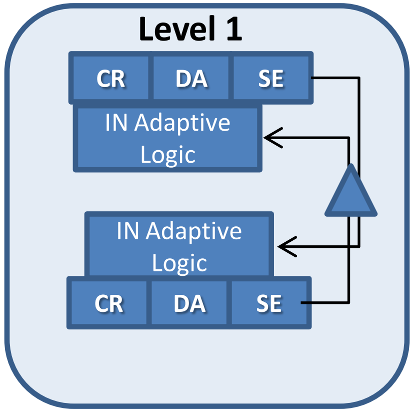
- Nodes help each other by sharing partial results
- Each Node may have different Domain KS (Partial Interpretations)
- Kernel subproblems insoluble at a Node without extensive Inter-Nodal communications
- Results achieved by one node influence or constrain those that can be achieved by another node

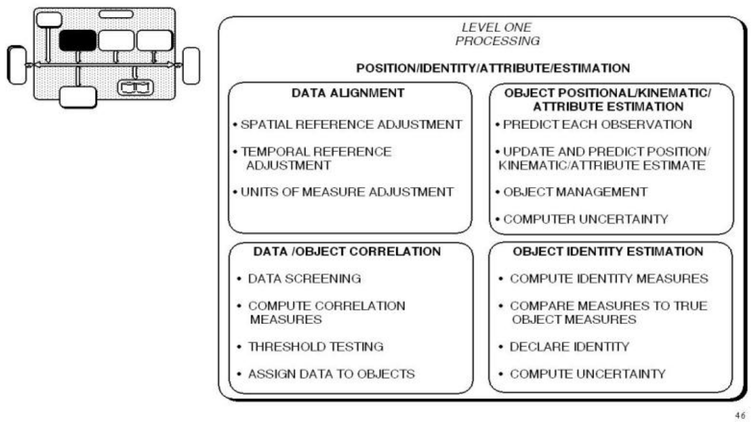
  --implicit partitioning ; higher communication reqmts

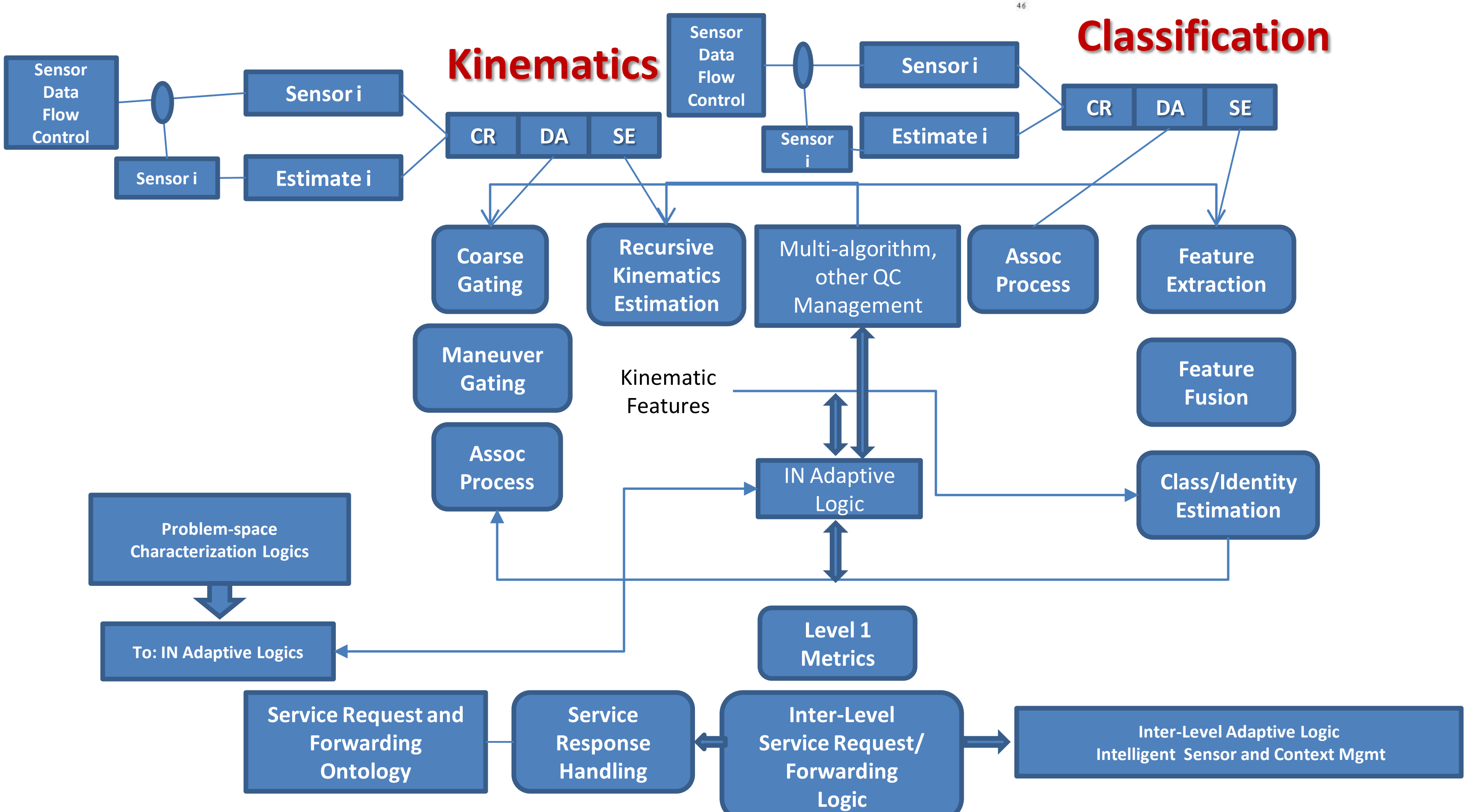
~ Reinforcement learning paradigm

A Generalizable Concept for all Levels??

4.2.1 LEVEL ONE
FUSION PROCESSING - OBJECT REFINEMENT







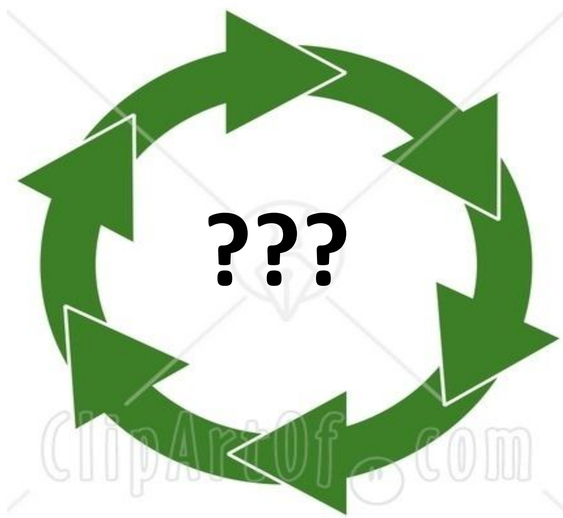
Possibly-applicable
Control-Theoretic Paradigms

Defining Applicable Control-Theoretic Concepts

SISO/MIMO

Stability

Applicability of PID concepts



Linear

Robustness

Non-linear

Nested Control*

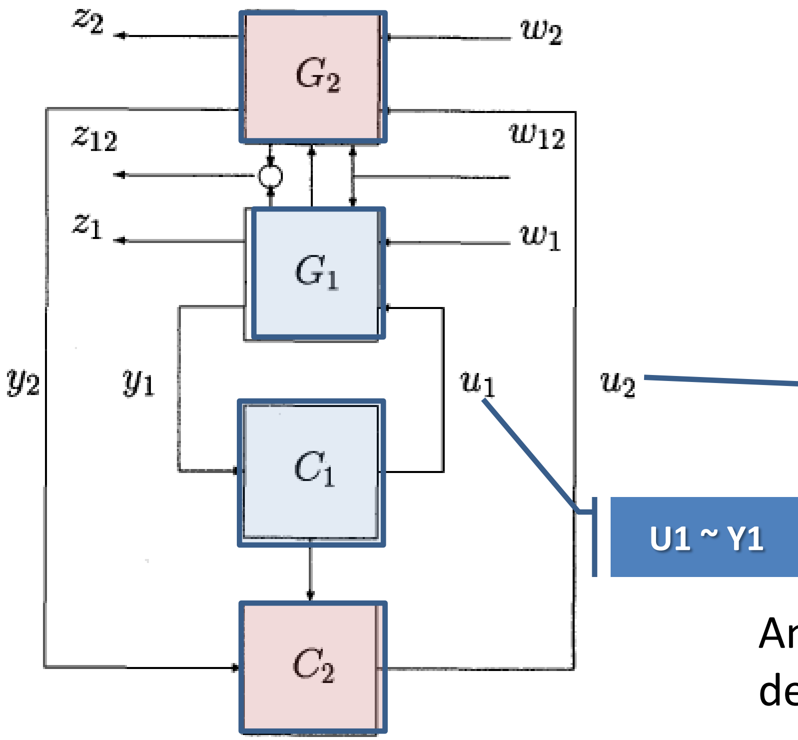


Figure 1: Nested Structure

G = plant/process
- G1 = internal subsystem
- G2 = external subsystem

C = controller
U = control inputs
Y = measured output
ω = external inputs
Z = outputs

Any control action at a particular nest (subsystem) depends only on the information of the subsystems inside the nest and not on any information outside of it.

Also, the control action at a particular nest (subsystem) does not affect the subsystems (nests) inside it but only the exterior ones.

Voulgaris, P.G., Control of Nested Systems, Proceedings of the American Control Conference Chicago, Illinois June 2000

Notional Nested Situational Estimation Processes

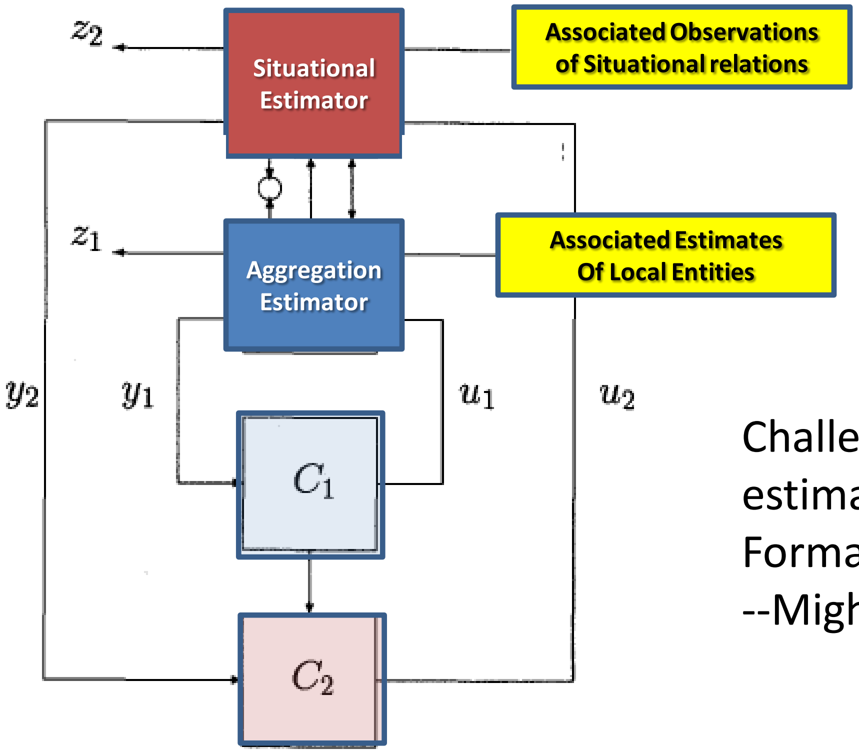
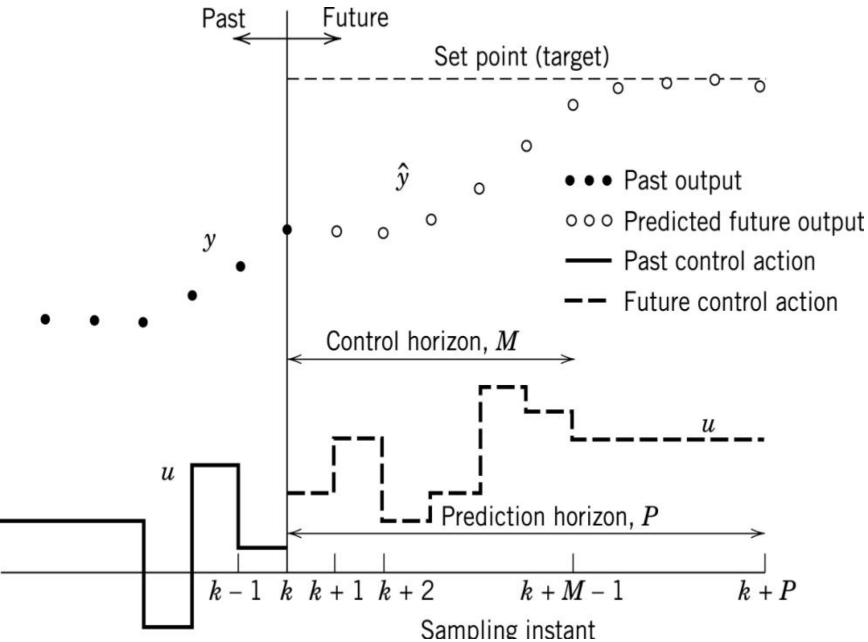


Figure 1: Nested Structure

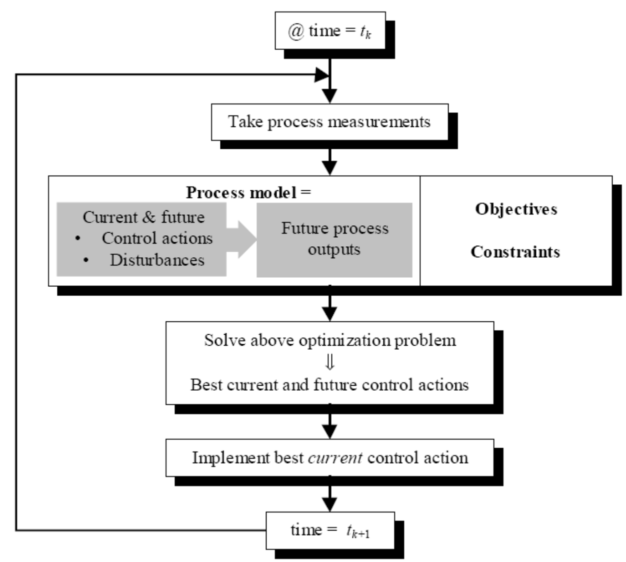
Y ~ quality metrics for each Estimation process
U ~ stopping criteria for each Estimation process
Z ~ State Estimates for each estimator

Challenge: Frame the Fusion cooperative estimation problem into a form that allows use of Formal control-theoretic solutions
--Might involve trading performance for stability

Model Predictive Control*

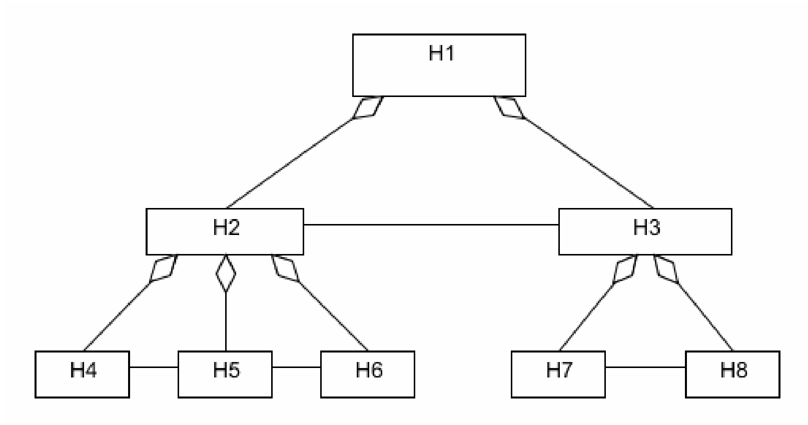


--Explicit use of a model to predict the Fusion process output along a future time horizon.
--Calculation of a control sequence to optimise a performance index.
--A receding horizon strategy, so that at each instant the horizon is moved towards the future, which involves the application of the first control signal of the sequence calculated at each step.



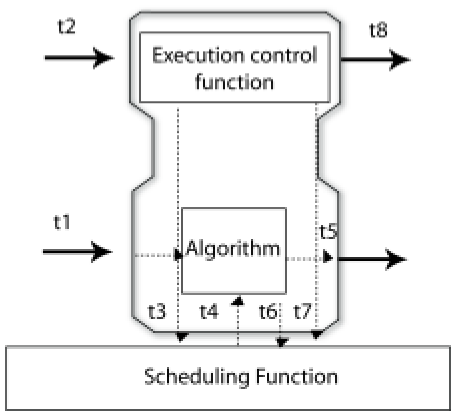
S. F. Page, , Adaptive Horizon Model Predictive Control based Sensor Management for Multi-Target Tracking Proceedings of the 2006 American Control Conference Minneapolis, Minnesota, USA, June 14-16, 2006

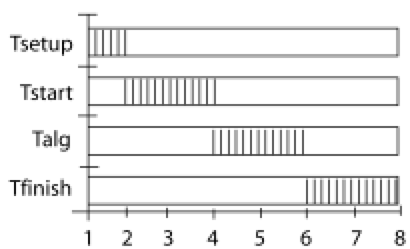
Holonic Control



Interconnected part-whole components

- See International Electrotechnical Commission[1] (IEC) Standard 61499, an open standard for distributed control and automation





“Execution Charts” but connection
To formal control theory unclear

P. McGuire eta al, The Application of Holonic Control to Tactical Sensor Management, Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS’06)

FIPA Agent Management Standard

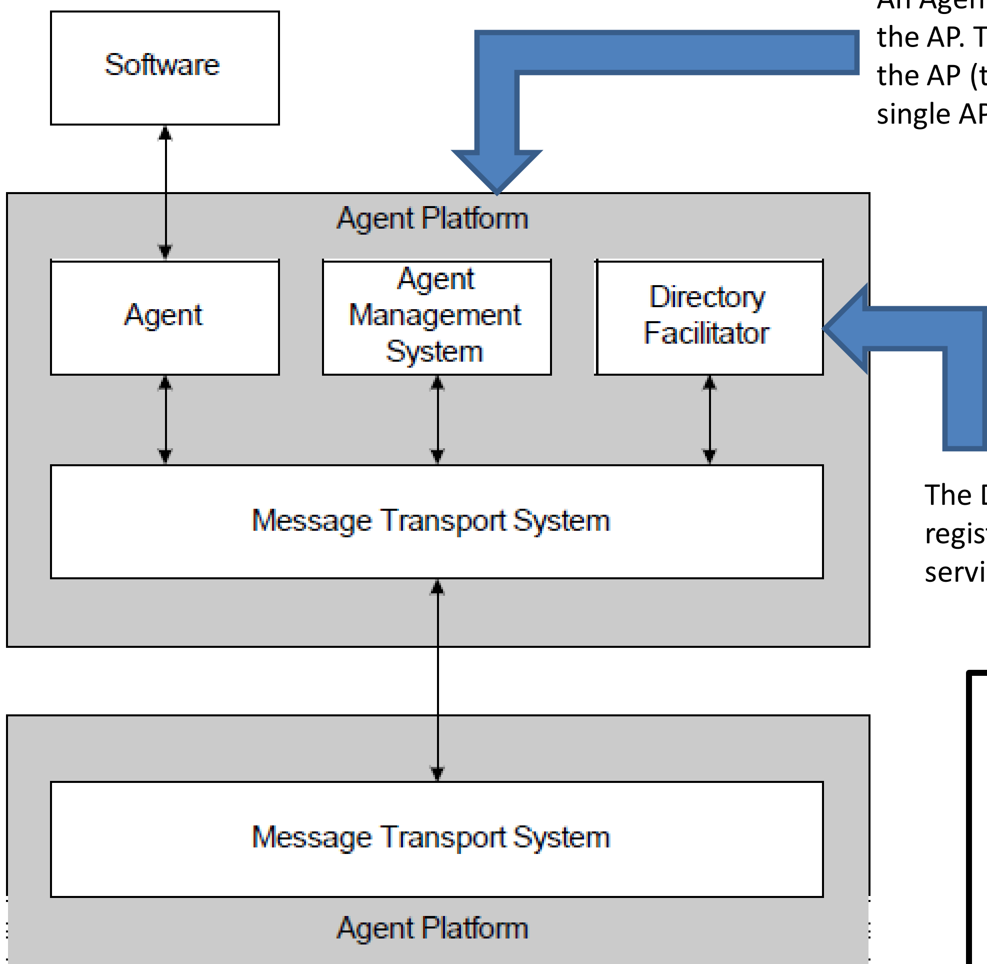


Figure 1: Agent Management Reference Model

An Agent Management System (AMS) is a mandatory component of the AP. The AMS exerts supervisory control over access to and use of the AP (that may contain multiple Agents). Only one AMS will exist in a single AP.

The DF provides yellow pages services to other agents. Agents may register their services with the DF or query the DF to find out what services are offered by other agents

It seems possible that a Results-Sharing approach could be achieved with an Agent paradigm

Blackboard Control

- General BB approach:
  - Sophisticated balance of opportunistic and goal directed control strategies
  - Intertwined with overall system design, e.g., number of KS´s affects control complexity and overhead
  - Many factors to consider* :
    - Extension and formalization of mechanisms for goal-directed control without loss of opportunistic control capabilities
    - Development of abstract models of the search space that can be used to make more accurate estimations of the long term global value of potential actions and to evaluate satisfaction of the termination criteria
    - The development of architectures that support the specification and application of explicit and sophisticated (highly context specific) control strategies
    - Concern with the efficiency of blackboard control

(see "The Evolution of Blackboard Control Architectures", Norman Carver and Victor Lesser CMPSCI Technical Report 92-71 October 1992)

Summary

- Achieving an implementable, reusable Fusion Framework has potentially high payoff for the IF community
  - It is worth the effort to study the feasibility of achieving such a Framework
  - But much needs to be done
  - Concepts presented here will be tested at UC3M
- It is an interesting research challenge problem
  - Interdisciplinary
  - Complex
  - Publishable, able to be prototyped and tested
  - Testing methods and metrics themselves a challenge

Characteristics of a Level 2 Approach

- Level 2 processing, fusion characteristics
  - Estimates (minimally L1) and measurements as input
    - Highly asynchronous
  - Partition SitEst problem to
    - Events
    - Behaviors
    - Aggregated Entities
    - Other TBD
    - Relations among above = Situational State

Or an Ontology of choice (Events through Relations among above)

Cooperating, synergistic, multiple Knowledge Sources (Events through Other TBD)

- Difficult to model dynamics and details; knowledge of varying confidence
- Requires some type of Opportunistic or Discovery based approach involving knowledge about categories of Entities and Relations
  - Multi-agent
  - Multi-Graphical
  - Blackboard with Multiple KS´s

Cooperative Distributed Problem-Solving*

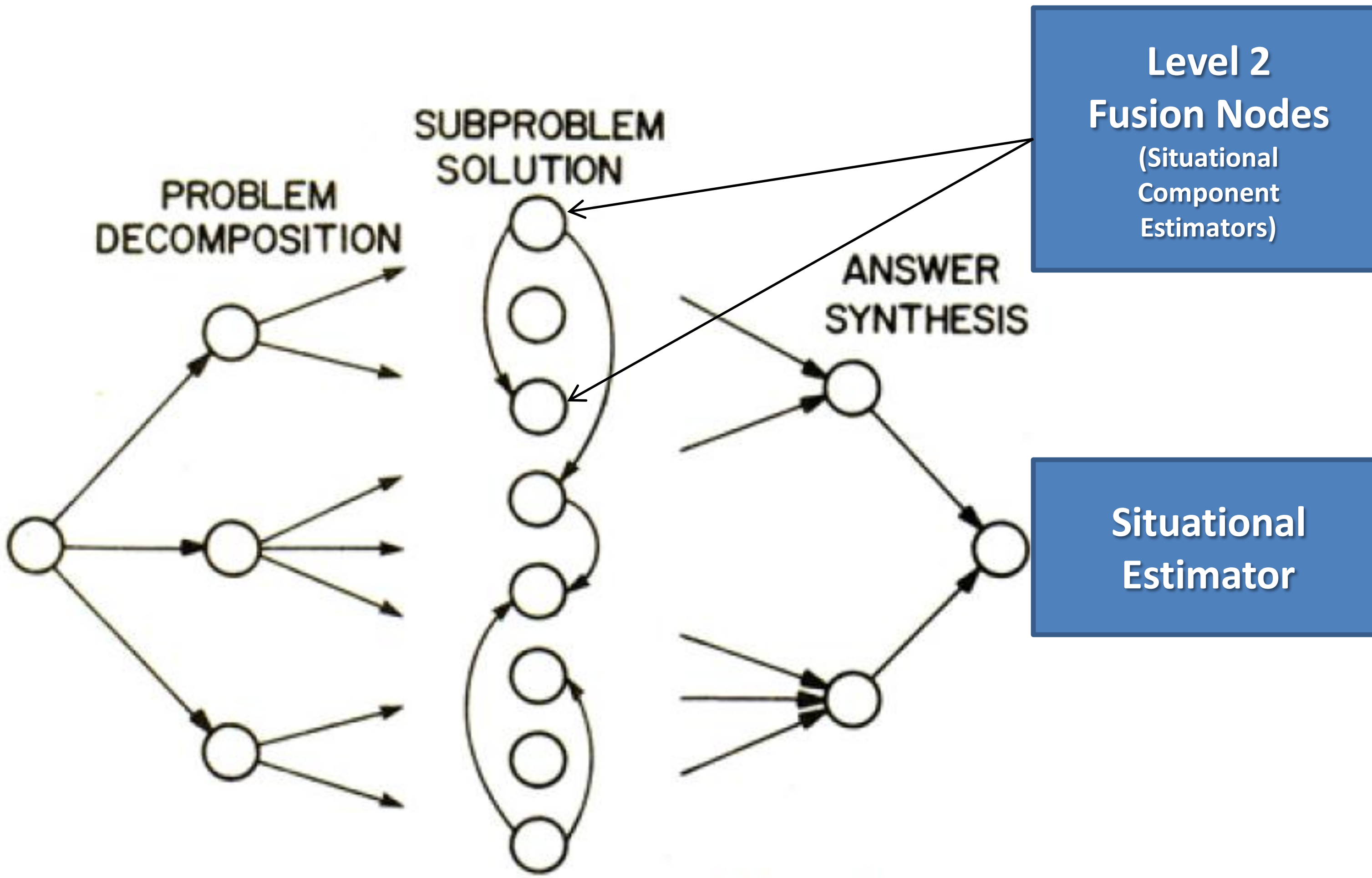


Fig. 1. Phases of distributed problem solving.

*IEEE Trans SMC, Vol. SMC-11, No. 1, Jan 1981 Frameworks for Cooperation in Distributed Problem Solving R. G. Smith, and R. Davis

Level 2 Fusion and the Blackboard Framework

- BB Problem Solving approach:
  - RESULTS-SHARING STRATEGY
  - incremental hypothesize and test or evidence aggregation; opportunistic
    - Hospitable to unpredictable situation evolution
  - Analogous to constraint-based techniques
    - Data constrains feasible hypotheses
  - Can be made “Greedy” by exploiting sub-hypotheses of lower uncertainty
  - Hospitable to multiple lines of reasoning to include alternative viewpoints (e.g., Red, Blue, etc)
  - Balances Goal-directed and Data-directed control

Blackboards and Control

- Why control if opportunistic?
  - Single processor constraints
  - Combinatorial applications (many KS´s)
    - Competition of computation of KS preconditions and KS inferencing generates high overhead, can delay convergence
- The task of the control component in a blackboard system is to determine which of the KS`s currently on the agenda has the *maximum expected value*
  - Actions that generate partial solutions
    - Involves solution uncertainty
  - Actions that develop better understanding
    - Involves control uncertainty

Agenda-based Control

- All possible actions are placed onto the agenda and on each cycle the actions are rated and the most highly rated action is chosen for execution
  - Some new KS action or new Domain Data changes Domain BB
  - Have to check KS preconditions (by type of BB event) to nominate next KS action
    - Precondition check (Precondition index)-initial KS candidacy
    - Triggering KS Preconditions (~LHS of Rule)-final KS candidacy/feasibility)
    - Rating (Value calculation)—and Ranking
    - Execution, if selected

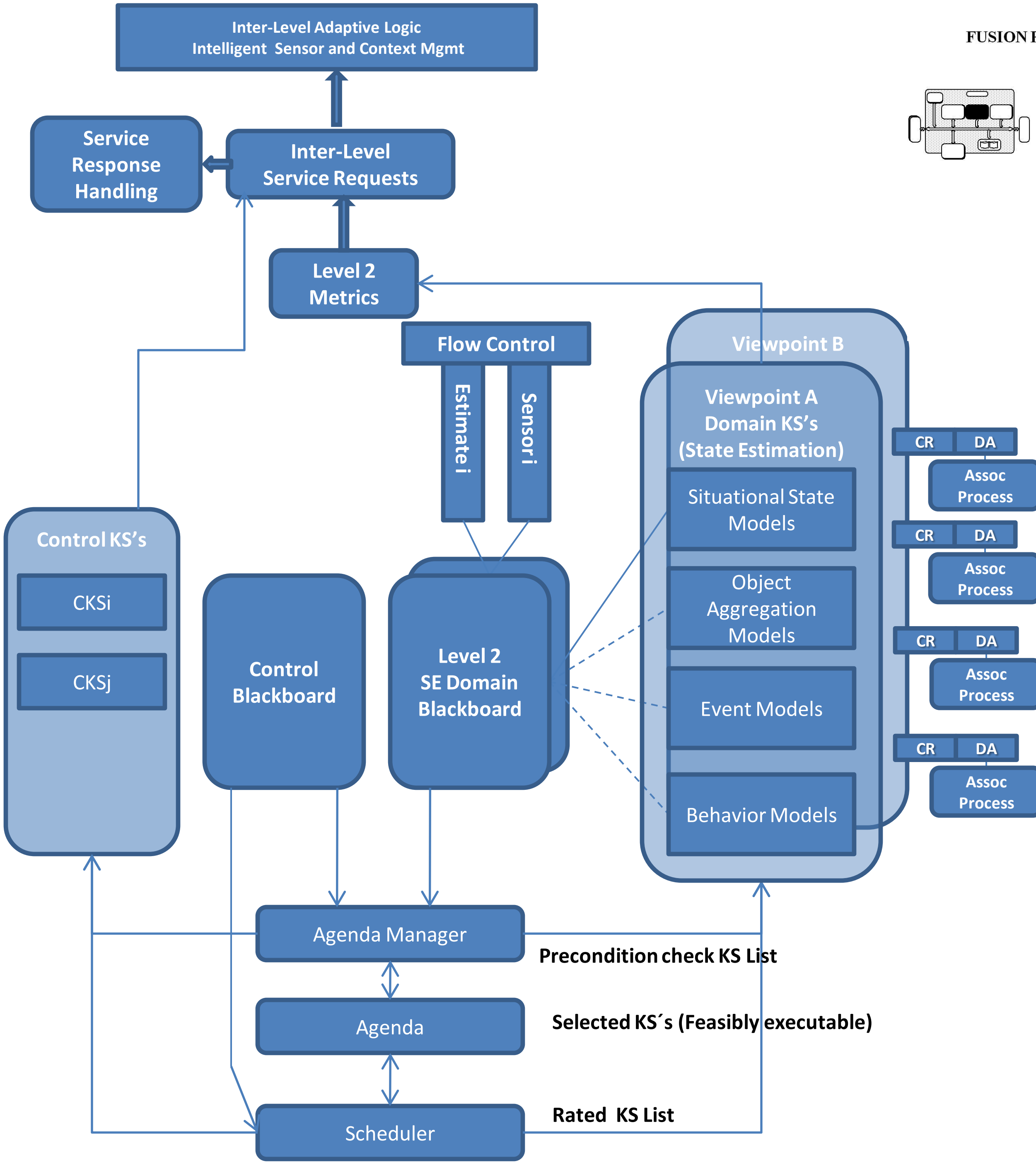
One possible BB candidate: BB1

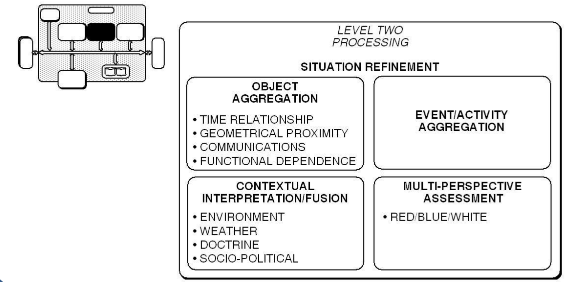
- Treats control problem as a BB problem
  - Has both a Control BB and a Domain BB
  - Method for doing comparative evaluations of possible actions
  - Allows adaptive control based on just-computed (updated) expected value
    - Dynamically changes rating functions
    - No single control paradigm needed for entire problem-solving process
  - Allows opportunistic control as well as opportunistic knowledge application
    - Achieving good balance can be difficult

BB1 as a Building-block

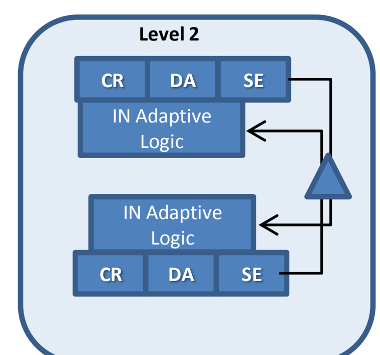
- Old (80´s) but still under development at Stanford
- Various versions available, both full system and kernel version (newer but incomplete version)
  - BBK is C++ version but is a Kernel
- Some reasonable starting documentation
- See http://www-ksl.stanford.edu/projects/BB1/bb1.html

4.2.2 LEVEL TWO
FUSION PROCESS - SITUATION REFINEMENT





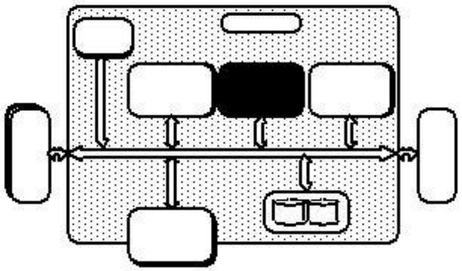
NB: Based on BB1 Approach



Way Ahead

- Further discussion within the Team
- Analysis to try fitting some UC3M applications to the Framework
  - Functional flow abstraction of UC3M application
  - Extract a simple, basic form of the Framework
- If successful, develop basic "α-version" of the software framework design
- Begin Test and Evaluation methodology definition

4.2.2 LEVEL TWO
FUSION PROCESS - SITUATION REFINEMENT



LEVEL TWO
PROCESSING

SITUATION REFINEMENT

OBJECT AGGREGATION

- TIME RELATIONSHIP
- GEOMETRICAL PROXIMITY
- COMMUNICATIONS
- FUNCTIONAL DEPENDENCE

EVENT/ACTIVITY AGGREGATION

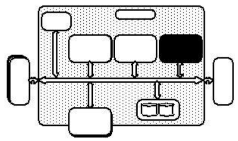
CONTEXTUAL INTERPRETATION/FUSION

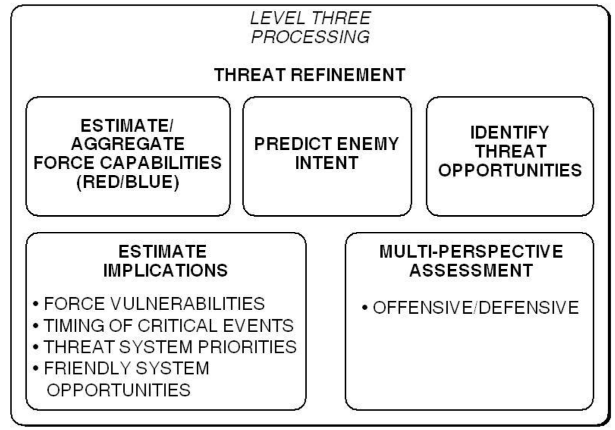
- ENVIRONMENT
- WEATHER
- DOCTRINE
- SOCIO-POLITICAL

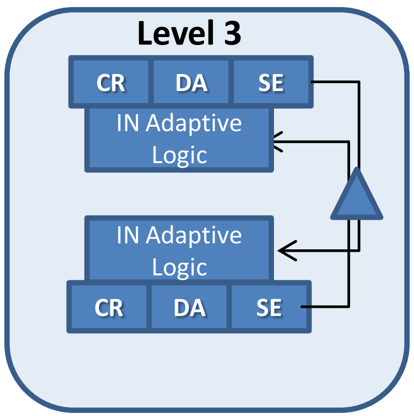
MULTI-PERSPECTIVE ASSESSMENT

- RED/BLUE/WHITE

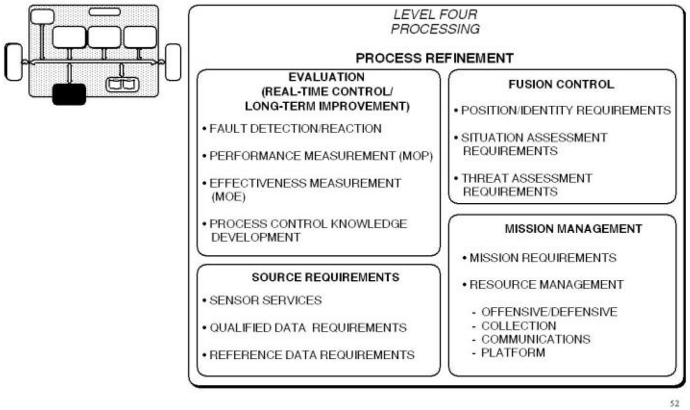
4.2.3 LEVEL THREE
FUSION PROCESS - THREAT REFINEMENT

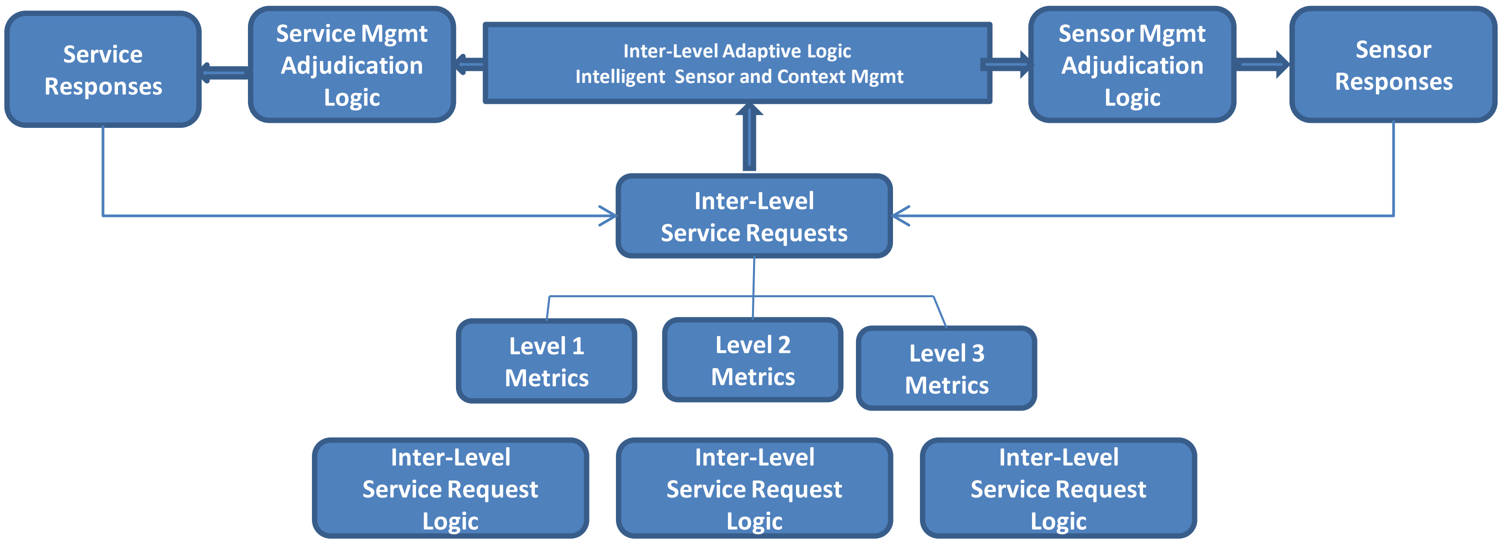






4.2.4 LEVEL FOUR
FUSION PROCESS - PROCESS REFINEMENT





Regarding Blackboard Architectures

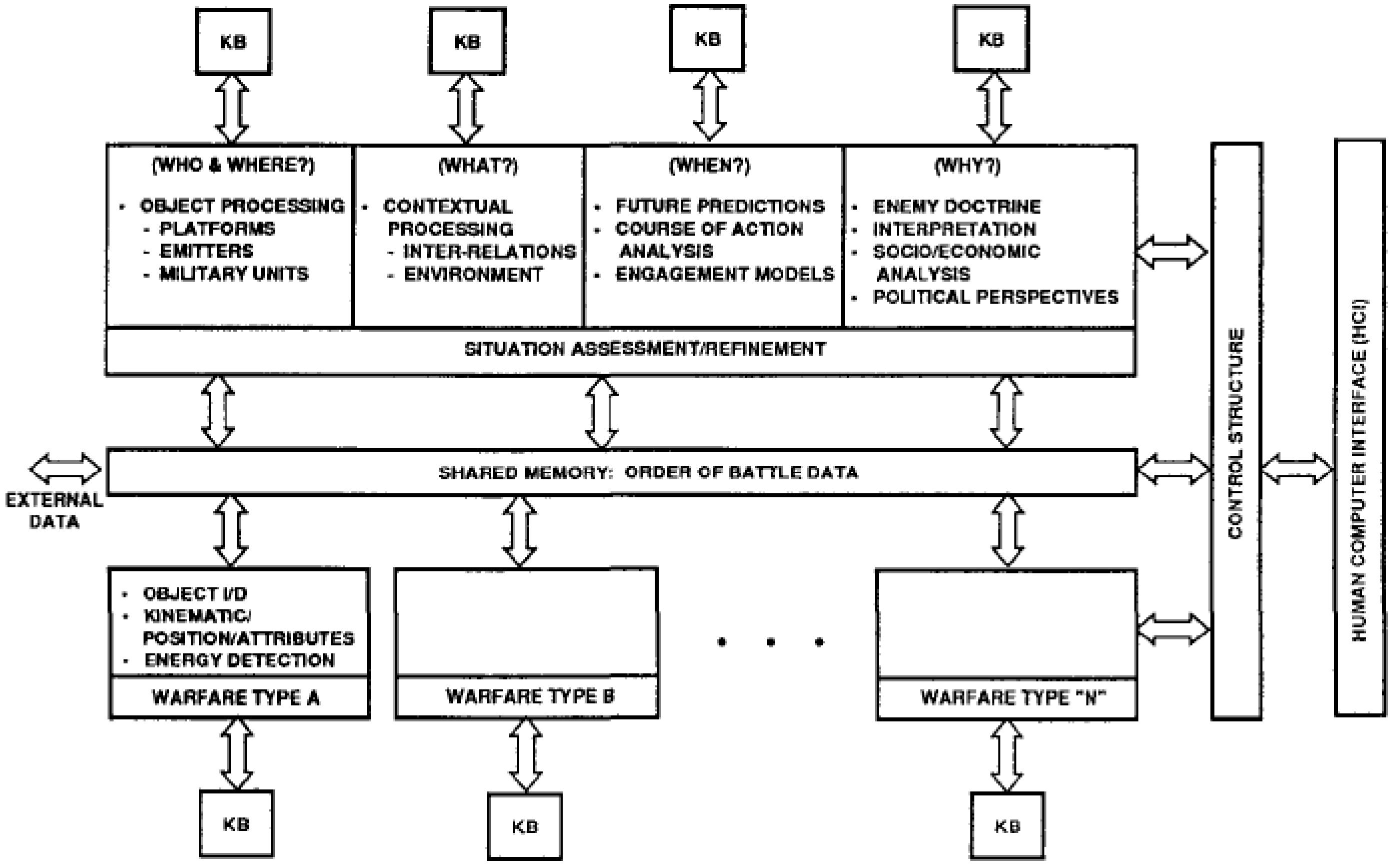


Fig. 3. Example of blackboard data fusion architecture.

Regarding Blackboard Architectures

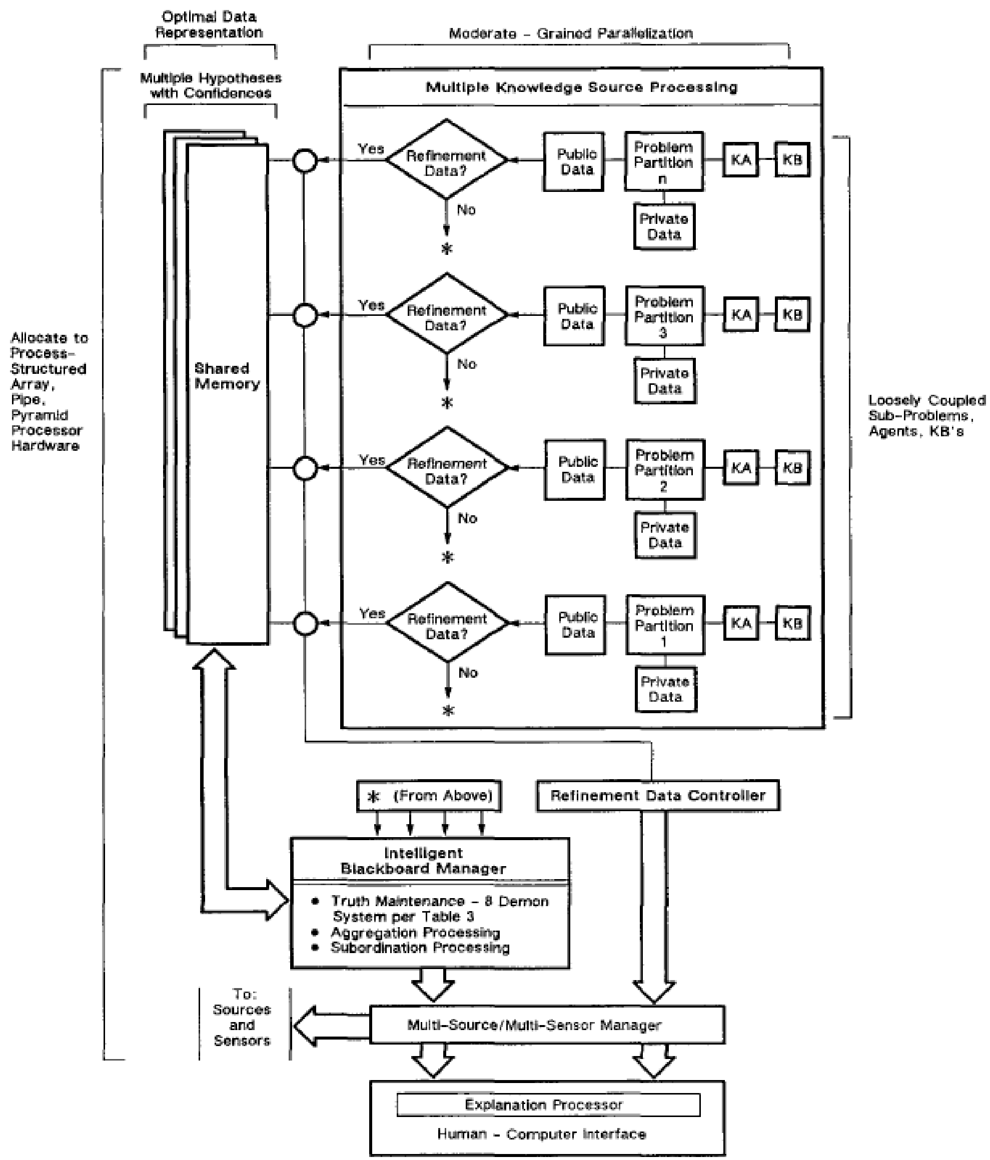


Fig. 4. Conceptually optimal blackboard architecture for data fusion applications.