# Real-Time Sensor Fusion Framework for Distributed Intelligent Sensors

**Xiaojing Yuan[1], Xiangshang Li[2], Xiaohui Yuan[1]**
1. EECS Department, 2. CEE Department, Tulane University, New Orleans, LA

**Abstract**

Multi-sensor data fusion has found widespread applications in industrial and research sectors. The purpose of real time multi-sensor data fusion is to dynamically estimate an improved system model from a set of different data sources, i.e., sensors. This paper presented a systematic and unified real time sensor fusion framework (RTSFF) based on distributed intelligent sensor network. The RTSFF is an open architecture which consists of four layers – the transaction layer, the process fusion layer, the control layer, and the planning layer. This paradigm facilitates distribution of intelligence to the sensor level and sharing of information among sensors, controllers, and other devices in the system. The transducer layer is populated with intelligent sensors. The learning ability of the intelligent sensor model enables it to extract characteristics of monitored signal. The representation issue is managed at this level. After describing the RTSFF, the paper then focuses on the fundamental units of the framework, the highly autonomous transducers.

## 1. Introduction

Multi-sensor data fusion has found more and more widespread applications in industrial and research sectors. The purpose of real-time multi-sensor data fusion is to dynamically estimate an improved system model from a set of different data sources, i.e., sensors.

The choice of architecture is a fundamental issue when designing a fusion system. Commonly used architectures include the three traditional architectures introduced by Hall [1], the centralized, the autonomous, and the hybrid. However, they did not give any directions on how the system should be designed to improve the efficiency of sensor fusion algorithms based on the relationship between units. The JDL fusion model, originated from the sensor fusion sub-panel of the US Joint Directors of Laboratories (JDL) [2], is helpful for common understanding of the functionalities usually involved when building a sensor fusion system. However, it is a data driven sensor fusion model and hard to be used as the basis for implementation. The OODA model [3] introduced the cyclic control cycle to represent the information gathering and decision making loop. However, it does not specify the sensor fusion tasks and which stage they belong in the structure. The omnibus sensor fusion model [4] tried to combine the sensor fusion functions defined in the JDL model and the OODA cyclic structure together. However, the omnibus model does not support hierarchical modular architecture that could partition tasks to support the distributed sensing and data processing. Thus, the model does not support reusable modules that can be implemented and tested separately for different applications. The waterfall fusion process model [5] separates different sensor fusion functionalities just as JDL model. Like JDL model, it does not provide means for feedbacks and actions either. It is notable that except for the JDL sensor fusion model, other models are not specifically designed for sensor fusion systems. They are created to model the decision making process at the beginning, but were used to demonstrate the functionalities, or the cyclic structure of the sensor fusion systems.

In this paper, we propose a real time sensor fusion framework (RTSFF), which includes detail information such as architecture, sensor fusion functions, and recommendation communication protocols. The framework is specifically designed for a distributed transducer network and supports hierarchical signal processing and sensor fusion functions in a real-time control system. However, it is open to accommodate different sensor fusion algorithms, new types of sensor and actuator, and other communication protocol standard. After describing all units in RTSFF, the paper gives detailed information on the fundamental units of the framework, the highly autonomous transducers (HATs). Then a cupola furnace example is used to demonstrate how a sensor fusion system is designed for distributed HATs for automatic monitoring and fault detection.

## 2. Real-Time Sensor Fusion Framework

Figure 1 shows the data flow in the hierarchical distributed RTSFF. It is the extension of the sensor fusion architecture introduced in [6]. It has four layers of data processing with well-defined interface between them. They are the HAT layer, the process fusion layer, the control application layer, and the planning and decision layer. Each of these four layers fulfills different objectives, and requires different information. In the figure, physical sensors and actuators are represented by circles and the abstract sensors and actuators are depicted as the box encompassing those circles. Interfaces between different layers are illustrated as integrated part of each layer. The *File Based Knowledge System (FBKS)* is depicted as cylinder (functioning as knowledge base). All other functions of the system are outlined as boxes.

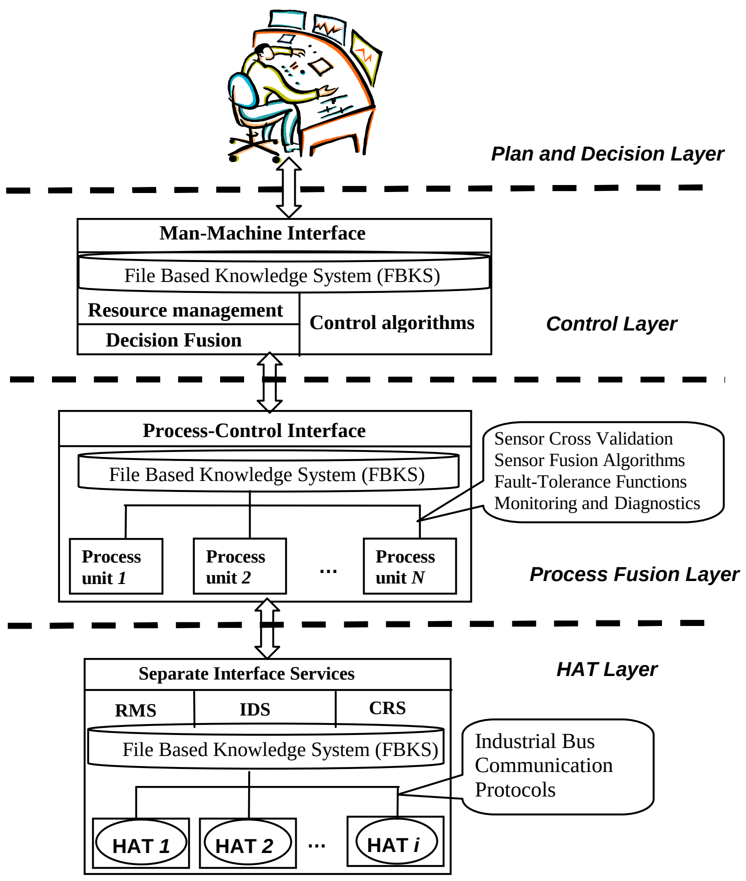


Figure 1 Real-Time Sensor Fusion Framework

## 2.1 Four Layers of the RTSFF

### 2.1.1 Highly Autonomous Transducer Layer

Each node in the HAT layer represents one or more physical transducers that interact with the environment directly. The task of a sensor is to observe a property of the environment, while the task of an actuator is to execute a control command. The HAT layer is populated with a network of these intelligent sensors and actuators. Except for its own health status, the intelligent actuators do not provide any more information to benefit the decision making process. Thus, in this paper, the focus will be on the highly autonomous sensors (HAS)[7]. The online model building and feature extraction functions in each HAS output unified qualitative behavior descriptors to the process fusion layer. Detailed functions for each HAS node is shown in Figure 2. The local sensor fusion algorithms look at the history of the signal, the information feed back form the process fusion layer, and the redundant readings if available to detect transducer-level failure and update the reliability of the HAS unit. In general, fusion of competitive sensing elements increases the robustness of the perception. Thus, when redundant readings are available, a fault-tolerance layer is added to increase the robustness of the unified view of the system.

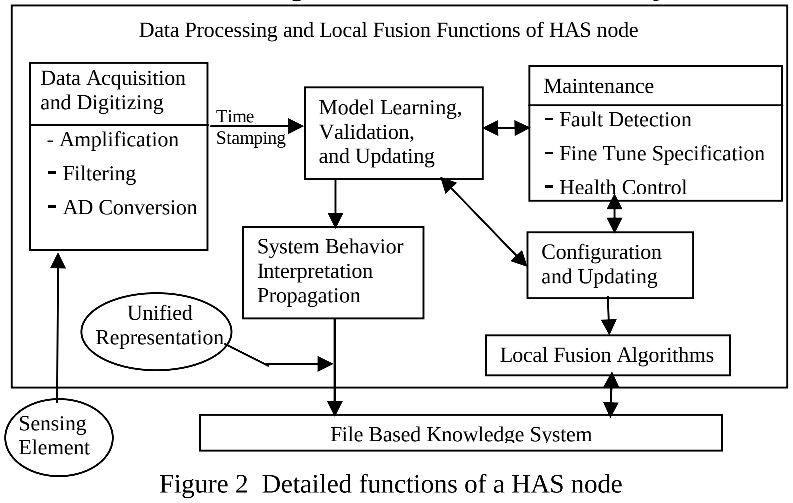

Data Processing and Local Fusion Functions of HAS node
Data Acquisition and Digitizing
- Amplification
- Filtering
- AD Conversion
Time Stamping
Model Learning, Validation, and Updating
Maintenance
- Fault Detection
- Fine Tune Specification
- Health Control
System Behavior Interpretation Propagation
Configuration and Updating
Unified Representation
Local Fusion Algorithms
Sensing Element
File Based Knowledge System

Figure 2 Detailed functions of a HAS node

### 2.1.2 Process Fusion Layer

The process fusion layer accommodates the hardware and software that act as glue between the HAT layer and the control application layer. It combines the measurements and qualitative descriptors from multiple HAS units into a more complete description of the environment, usually in the form of a map or matrix. In general, fusion of observations from complementary or cooperative sensors provides an extended and more complete view. It supports more reliable interpretation of the system behavior, and therefore supports more accurate diagnosis when abnormal behavior occurs. The process fusion layer detects the system-level failure based on fusion results, and then triggers the diagnostic functions. Depending on the application, the relationship between involved HAS units, and the available resources such as processing power, working memory, etc., different sensor fusion algorithms can be used in the process fusion layer. In the cupola furnace example (section 4), to fuse the numerical values from different HAS units, weighted confidence averaging is used. The fused value is then fed back to the HAS unit to update the model for behavior interpretation. To fuse qualitative descriptors from HAS units, a map is built to show the reliability of each HAS unit and the output of the process is also monitored to show the health status of the process as a whole.

### 2.1.3 Control Application Layer

The control application layer receives specifications from the operator and makes decisions about the control strategy based on the working condition of the whole system. It initiates actions in order to achieve a given goal, monitors and manages all process fusion units, and adjusts objectives according to the changes in whole system and the environment. The decision fusion at the control layer combines decisions and views from associated process fusion units. The fused results will be used to facilitate the resource management, the prediction of the maintenance time, and the adjustment of the control strategy.

### 2.1.4 Plan and Decision Layer

The plan and decision layer interact with human operator and makes decisions on the overall goal of the system, such as the maximum yield. The focus of the design is on the human computer interaction (HCI) at this layer. The basic rule is to provide accurate and sufficient information, while hide any unnecessary details, so that operators can make good decisions. The HCI should also allow the operators access any detail information he/she thinks necessary. The hierarchical maps in the RTSFF, i.e., the HAT-level map, the process-level map, and the system-level map, gives the operator the flexibility to access any information he/she needs.

#### 2.1.5 Fieldbus as Communication Protocol

Fieldbus technology drastically changed the industrial systems and is gradually replacing the 4-20 mA analog transmission that most field devices currently employ. After comparing several local area network protocol dedicated to industrial automation, which usually are proprietary protocol, we select FOUNDATION Fieldbus [8] as the communication protocol between each units in RTSFF. FOUNDATION Fieldbus is an open, interoperable Fieldbus that supports multi-drop, bi-directional, digital communication. It allows truly distributed control of the field devices. In summary, the FOUNDATION Fieldbus protocol can be used to connect all kinds of instrumentation systems, electrical devices, and analyzers from different manufacturers and integrate them into one system.

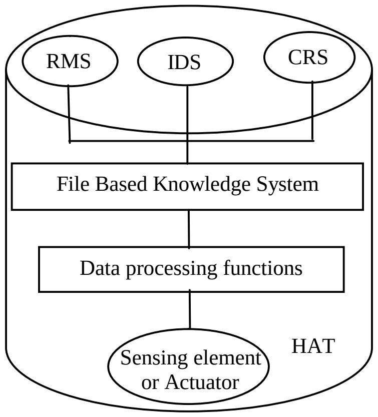
## 2.2 File Based Knowledge System

All information generated or shared among nodes in the RTSFF is stored in the File Based Knowledge System (FBKS). It can be implemented as centralized knowledge system, or hybrid system where each node has partial information embedded. The unified addressing scheme allows access to any information in any node from any node. For the real-time control system, it is important to separate the communication services between different layers so that the different information, such as sensor readings, configurations, internal state reports, and self-describing data, can be communicated in the correct format at the correct time interval.

### 2.2.1 Interfaces and Communication Service Separation

Three interfaces are designed to deal with communication between four layers. They are HAT-process interface, process-control interface, and man-machine interface. These interfaces decouple the communication activities from the local functions. The HAT-process interface provides three communication services to support sensor fusion algorithms for different purpose, such as real-time maintenance, online diagnosis, or dynamic configuration and plug-and-play setup. The process-control interface negotiates between the process fusion layer and the control application layer. It provides bi-directional accessibility to FBKS at each process fusion unit and combines maps from all process fusion units into one overall system map. Through the interface, the control layer sends commands and parameters to FBKS at each process fusion unit, and access overall system map generated by the interface. The man-machine interface between control application layer and plan and decision layer provides customized view of the system for different user groups. It also enables operators to define the objective of the control system, manually adjust parameters for each unit. It has to be carefully designed to reach the balance between not enough information to support decision making and too much detailed information that makes it impossible to reach any decisions.

Figure 3 shows the communication service separation in the HAT-process interface. Information in FBKS can be accessed via three communication services: real-time monitoring service (RMS), interpretation and diagnosis service (IDS), and configuration and resources management service (CRS). The RMS provides periodic communication at predictable time to transmit information for normal control loop at a pre-defined frequency. The fused results and maintenance information also fed back from the process fusion layer through RMS at pre-defined time interval. The IDS, on the other hand, provides communication at a much lower pre-defined frequency to transmit updates of the health status and other diagnostic information, or at sporadic time when diagnosis function is triggered, provided that it does not interfere with the timing of communication through RMS. The CRS provides bi-directional communication between the control algorithm and a particular HAT node that allows reading and modifying data at the node. Parameters and other initial information are transmitted to newly connect or reallocated HAT unit through CRS at sporadic time. In summary, the RMS has the highest priority and must not be interfered by IDS and CRS. The CRS has the lowest priority and will be scheduled when communication traffic is low. The separation of these three services allows online maintenance, thus improves the productivity of the system.


RMS
IDS
CRS
File Based Knowledge System
Data processing functions
Sensing element or Actuator
HAT
Figure 3 Communication Services Separation at HAT node

### 2.2.2 File Based Knowledge System

The file based knowledge system (FBKS) supports information sharing among different components of the system. The FBKS encapsulated in each node stores all data/information it generates and that requested and received from other nodes. The uniform naming and addressing scheme of FBKS enables it to function as both the sink and the source for all information transmission. Thus supports decoupling of the communication activities from local functions at each node. This ensures that each component of the system can function independently, thus supports truly distributed control system.

Figure 4 shows the structure of the uniform naming and addressing scheme of the FBKS. The current configuration allows maximum 256 processes in one system, and 256 HATs in one process unit. Each HAT unit can store up to 32 files of records, in which 28 of them stores data and descriptors at different abstract levels and the remaining four files are reserved for intermediate storage purpose. Each files stores up to 1024 records. The observation can occupy 32 or 64 bits according to different communication protocol. For complex system, the FBKS can be easily extended to accommodate more HATs and process units.

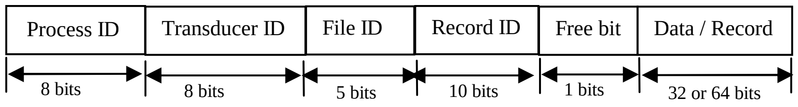


Figure 4 Naming and Addressing scheme of FBKS

### 2.2.3 Clock Synchronizing

In order to align all observation and measurements and provide a base for all sensor fusion algorithms and other data processing function, the RTSFF uses a global notion of time for each node in the distributed system. Thus, it is crucial to synchronize all clocks in the nodes to a global time base. The global time base is provided by the clock of a master node, which has an oscillator with very little drift rate. The clocks in other nodes use less costly on-chip oscillators. They are synchronized periodically to the master node's clock by checking the synchronize pattern at the end of every message from the master node. The typical clock drift rate is $10^{-3}$s.

## 2.3 Qualitative Descriptor Extraction for Highly Autonomous Transducers

Model learning and behavior interpretation function in each HAT node provides a basis for local fusion functions, fault detection and diagnosis, and configuration updating. In this section, the qualitative descriptors used to describe and infer the behavior of the system are described.

In the HAT model, learning is defined as identifying new concepts and behaviors and save them in the FBKS. Since concept and behaviors are qualitative descriptors describing the system conditions at different abstract level, the online learning function is denominated as Qualitative Engine (QE).

Figure 5 shows the general scheme for concepts and behaviors extraction. Qualitative descriptor, "Behavior" is defined by a succession of "Concepts" which are, in turn, defined by "Properties" that maintain constant values for a number of samples of the signal being read by the sensor. Table I shows how one may determine the concepts that are needed to specify a behavior. For example, given the behavior *step change*, the related concepts defined by the experts are constant, noise, sudden jump. Each of these concepts has to be defined in terms of a set of properties, their values at every sample time, and the duration of the properties remain the same.

The online learning capability allows HAS to pick up unpredictable and previously unnoticed concepts and behaviors on the fly. Thus, it won't miss any new scenarios exhibited by the system, for instance, when it experience a change of workload.

During the HAS maintenance period, more sophisticated offline modeling and learning algorithms can be used to fine tune and update the underlying local system model.

Based on the knowledge learned form the online and offline learning process, self-validation and diagnostic function are embedded into HAT.

A set of functions in the QE, $F=\{f_1(x), f_2(x),...,f_s(x)\}$, maps the sensor readings in the input space ($\Re$) to feature space ($Q$ $\{q_1, q_2,...,q_s\}$). Then QE learns the concept c defined by a set of qualitative features $\{q_1, q_2,...,q_s\}$. A behavior descriptor, $b$, learned by *QE*, is defined by a set of consequent concepts, $\{c_1, c_2,..., c_q\}$. When learning concepts, if we define s qualitative properties (*q*), each has *l* possible states, and then the number of all possible combinations of the elements of $q_s$ that defines concept C is $(s*l)^s$. For a property set of five, each with three possible states, the number of all possible combinations is 759375. This number will increase exponentially as the number of properties increases. However, not all these combinations are physically feasible. Thus, it is unwise if not impossible to create a knowledge base with all possible patterns beforehand. The same consideration is true for the behavior even if we only deal with fixed number of concepts per behavior. The online learning ability of QE enables it to learn new concepts and behaviors whenever they occur. Thus, only the concepts and behaviors that are

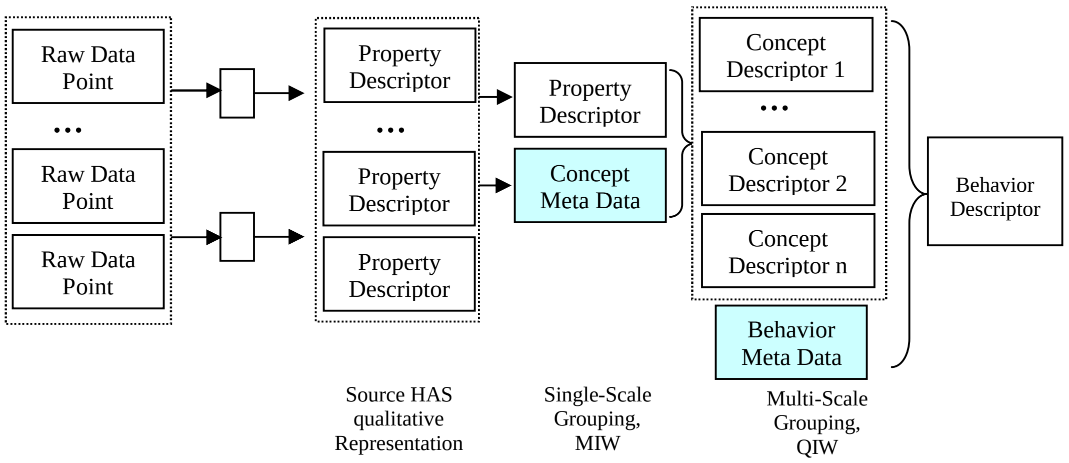


Figure 5 General Scheme for Property, Concept, and Behavior Extraction

Table I Measurand and Sensor Behaviors and related concepts

| SITUATION | DESCRIPTION | ASSOCIATED CONCEPTS |
|---|---|---|
| Constant input | Except for noise, the output remains at a constant level during the monitoring time. | Noise, monitoring time, constant level. |
| Monotonic change | Except for noise, the output increases or decreases linearly with time during the monitoring time | Noise, linear increase or decrease, monitoring time |
| Step input | Except for noise, the output remains constant during the monitoring time, then it suddenly jumps to a higher or lower level, where it remains during the monitoring time. | Sudden jump, noise, constant, monitoring time. |
| Harmonic | Except for noise, during the monitoring time, the output changes at a certain principal frequency. Other secondary frequencies may also be present. | Frequency, principal frequency, secondary frequencies, noise. |
| Spike | Random occurrence of a large amplitude pulse with low energy. | Random occurrence, short duration, large amplitude, low energy |
| Excessive Noise | Increase in the noise level at random time with low or high energy, lasting some time. | Random occurrence, noise level, longer duration, low or high energy level. |
| Zero Value | The output value is stuck at zero. | Zero value, no noise, no energy. |
| Stuck value | The output value is stuck at some number. | Constant value, no noise, low or high energy. |
| Drift | A small monotonic change of the signal from the actual value over a long period of time. | Small monotonic change, long duration time. |

physically feasible are stored in the knowledge base (FBKS). The result is a significant decrease in space complexity of the knowledge base and consequently, reduced time complexity for searching and matching algorithms.

## 3. Simulated Results: Fault Detection of Cupola Furnace

The Cupola iron furnace is an essential part of most cast iron foundries [9]. The diameter of a Cupola furnace ranges from two to 15 feet, and is usually charged with coke, metal and other materials such as enzymes. Figure 6(a) shows a model for Cupola furnace with three pairs of temperature and pressure sensors located at the bottom, the middle, and the top of the furnace. From top to bottom, Figure 6(b) shows the iron temperature measured by $T_1$, $T_2$, and $T_3$ respectively. It also shows the manually labeled behaviors of the signal. Besides the normal change of load, which results in the step changes in the iron temperature, noises such as spikes and disturbances are also treated as behaviors here.

The three pairs of complementary and cooperative sensors monitor the same parameter at different location, thus provide more comprehensive information about the conditions within the furnace. Because of the location of these temperature sensors, there are noticeable delay in their readings, and different level of noises. The delay is approximated by comparing the response time when working condition changes, i.e., starting point of step changes. The time lag between $T_1$ and $T_2$ is 25 points, and the time lag between $T_2$ and $T_3$ is the same.

Using the highly autonomous sensor model described before, all behaviors of each signal can be detected, including the normal behavior "step up" and the sensor errors, such as "disturbance" and "spike". When these errors are detected, the fused values, based on the fusion of three complementary temperature sensors, will be used for decision making of control strategy and the confidence of the sensors will be reduced accordingly. The detailed algorithm to adjust the confidence and numerical fusion is described in [10]. The detected concepts and behaviors are shown in table II. Notice that the format of each record follows the uniform naming and addressing scheme of FBKS. Figure 7 explains the elements of the "step up" behavior record.

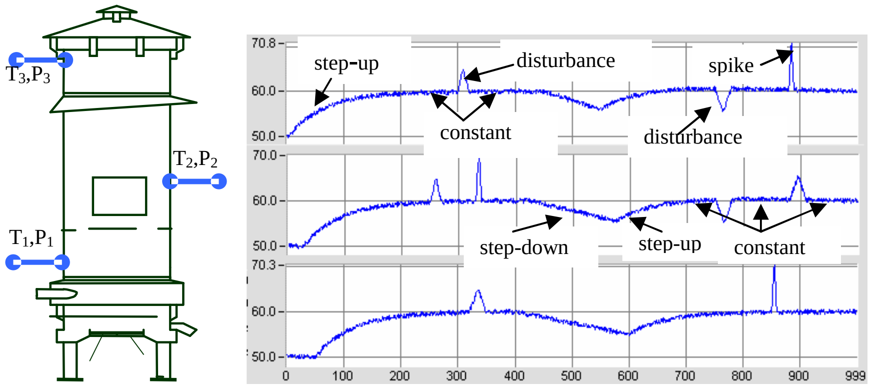


Figure 6 (a) Cupola furnace and (b) The simulated iron temperature under different working condition

## 4.Conclusion

In this paper, we introduced a unique, open structure real-time sensor fusion framework (RTSFF) and the underlying highly autonomous sensor (HAS). The advantages of the RTSFF include: open structure that can accommodate different sensor validation and fusion algorithms; communication separation enable most of the diagnosis, maintenance, and configuration functions to be performed online without interfering with the real-time data transmission under normal working condition; the uniform naming and addressing scheme of the file based knowledge system (FBKS) decouples the communication activities from the local functions; the master-slave clock synchronization mechanism fully utilize the capability of the Fieldbus communication protocol and reduces the cost for implementation of real-time sensor fusion systems. The learning ability of the HAS not only provides a uniform qualitative representation for higher level decision fusion, but also facilitates the health monitoring of the sensor itself, thus guarantee the input to the fusion algorithms will not be garbage. The Copula furnace example demonstrate how the numerical fusion of three complementary temperature sensors help the detection of disturbances and errors of the sensor, and how qualitative descriptors are extracted by the HAS units and stored in the FBKS. The HAS and RTSFF can be used for monitoring the real-time systems with time series signals, as well as the real-time systems where images are the output product. The unified qualitative representation of concepts and behaviors can be treated as patterns or objects in the images. This is very useful in Ladar and Lidar applicaitons, when information at both one and two dimension need to be fused together at decision level.

## References

[1] Hall, D.L., 1992. *Mathematical Techniques in Multi-sensor Data Fusion*, Mass.: Artech House.

[2] Steinberg, A.N., Bowman, C.L., and White, F.E., Oct.1998. *Revision to the JDL data fusion model*, Proc. 3rd NATO/IRIS Symposium, Quebec, Canada.

Engelmore, R., and Morgan, A. eds. 1986. *Blackboard Systems.* Reading, Mass.: Addison-Wesley.

[3] Boyd, J.R., May 1987. *A Discourse on Winning and Losing*. Unpublished set of briefing slides, Air University Library, Maxwell AFB, AL, USA.

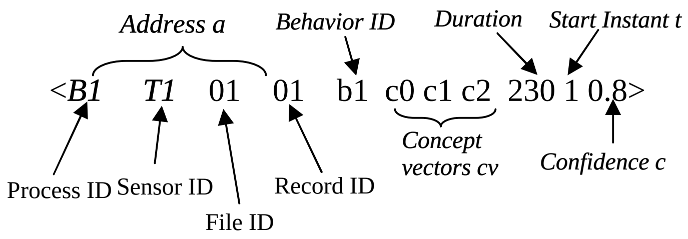


Figure 7 Record for behavior *Step Change Up*

[4] Bedworth, M.D., O'Brien, J., July 1999. *The Omnibus Model: A New Architecture for Data Fusion?* FUSION'99, Helsinki, Finnland.

[5] Markin, M., Harris, C., Bernhardt, M., Austin, J., Bedworth, M., Greenway, P., Johnston, R., Little, A., and Lowe, D., 1997. *Technology Foresight on Data Fusion and Data Processing.* The Royal Aeronautical Society.

[6] Yuan, X. Figueroa, F., Nov. 2001. *Intuitive Intelligent Sensor Fusion with Highly Autonomous Sensors*. Symposium on Instrumentation, Measurements, and Sensors, DSC, IMECE2001.

[7] Figueroa, F., Yuan, X., 2000, *A Taxonomy and Environment to model any sensor as highly autonomous sensor*, International Workshop on Virtual and Intelligent Measurement Systems, Annapolis, MD, USA.

[8]Yokogawa Electric Corporation, 2001. *Fieldbus Technical Information*, T1 38K03A01-01E.

[9] Frolik, J., Abdelrahman, M., Kandasamy, P., 2001. *A confidence-based approach to the self-validation, fusion, and reconstruction of quasi-redundant sensor data*, IEEE Transactions on Instrumentation and Measurements.

[10] Yuan, X., Li, X., Buckles, B., 2004. *Real-time sensor validation and fusion for distributed autonomous sensors*. Submitted to SPIE2004.

Table II Concepts and Behaviors Extracted from Sensor Readings of $T_1$

| | $T_1$ | Description |
|---|---|---|
| *Behavior Descriptors* | *<B1 T1 01 01 b1 c0 c1 c0 277 1 1.0>* | b(1): Step Change Up from 1-278, conf(1.0). |
| | *<B1 T1 01 02 b2 c0 c1 c2 c1 76 278 1.0>* | b(2): Pulse Disturb from 279-354, conf(1.0). |
| | *<B1 T1 01 03 b3 c1 c2 c1 207 279 1.0>* | b(3): Step Change Down from 355-561, conf(0.9). |
| | *<B1 T1 01 04 b4 c1 c0 c1 152 562 0.9>* | b(4): Step Change Up from 562-714, conf(0.9). |
| | *<B1 T1 01 05 b5 c3 c2 c1 c0 71 715 0.9>* | b(5): Pulse Disturb from 715-786, conf(0.9). |
| | *<B1 T1 01 06 b6 c1 c0 96 787 0.8>* | b(6): Constant to Up fro 787-880, conf(0.8). |
| | *<B1 T1 01 07 b7 c4 c5 c0 c1 18 881 0.8>* | b(7): Spike from 881-899, conf(0.8). |
| | *<B1 T1 01 08 b8 c2 c1 140 746 0.7>* | b(8): Down to Constant from 900-1000, conf(0.7). |
| *Concept Descriptors* | *<B1 T1 01 32 c0 nnnngsvv1+23 1 1.0>* | c(0): Upward trend with noise |
| | *<B1 T1 01 33 c1 nnnngsvv1+1;_ 110 1.0>* | c(1): Constant with noise |
| | *<B1 T1 01 34 c2 nnnngsvv1-2>} 280 1.0>* | c(2): Downward trend with noise |
| | *<B1 T1 01 35 c3 nlnngsvv1-2=} 500 1.0>* | c(3): Downward trend with low amplitude |
| | *<B1 T1 01 36 c4 nohmbpii1+2<\| 881 1.0>* | c(4): Steep Upward trend, sensor domain |
| | *<B1 T1 01 37 c5 nohmbpvv1+2;\| 883 1.0>* | c(5): Steep Upward trend, back to measurand domain |
| | *<B1 T1 01 38 c6 nlnngsvv1+1<_ 911 1.0>* | c(6): Low constant with noise |