

# A General Data Fusion Architecture

**Hervaldo S. Carvalho**

Center For Future Health  
University of Rochester  
Rochester, NY, U.S.A.

hervaldo\_carvalho@futurehealth.rochester.edu

**Wendi B. Heinzelman**

Electrical and Comp. Eng. Dept.  
University of Rochester  
Rochester, NY, U.S.A.

wheinzeln@ece.rochester.edu

**Amy L. Murphy**

Computer Science Dept.  
University of Rochester  
Rochester, NY, U.S.A.

murphy@cs.rochester.edu

**Claudionor J. N. Coelho**

Dept. of Computer Science  
Federal University of Minas Gerais  
Belo Horizonte, MG, Brazil

coelho@dcc.ufmg.br

**Abstract:** *Data fusion is an important component of applications for systems that use correlated data from multiple sources to determine the state of a system. As the state of the system being monitored and available resources change, the general data fusion framework should change dynamically based on the current environment and available resources in the system. To achieve this goal, we have proposed a general Data Fusion Architecture (DFA) based on the Unified Modeling Language (UML) and using a taxonomy based on the definitions of raw data and variables or tasks. The DFA can be reconfigured according to the measured environment and availability of the sensing units or data sources, providing a graceful degradation in the view of the environment as resources change. We have shown that we can apply the DFA to different domains and applications, including a test bed health monitoring application.*

**Keywords:** Data fusion architecture, adaptation, UML, resource management.

## 1. Introduction

We have developed a data fusion architecture that can be applied for network-based dynamic distributed systems or conventional stand alone centralized systems. The following characteristics can co-exist in both systems: components of the system can be inserted or excluded without stopping the entire system; the environment changes with time, as do the physical measures from it; an application's necessity changes according to different states of the environment, and as a consequence, it can use or reuse components in the different states; the system can be mobile and battery powered, so each component of the system has a different lifetime, bandwidth usage and power consumption. In the case of a network-based system, bandwidth is limited. All of these problems should be addressed from the application perspective to make the system robust to the dynamic environment. We think that some of these problems can be addressed at the data fusion module of the application.

Data fusion is a formal framework used to express the convergence of data from different sources in which is expressed the means and tools for the alliance of data that originated from different sources [1]. The US Department of

Defense has defined data fusion as a multilevel, multifaceted process dealing with the automatic detection, association, correlation, estimation, and combination of data and information from single and multiple sources [2]. The resulting information is more satisfactory to the user when fusion is performed than simply delivering the raw data [3]. In data fusion, information may be of various types, ranging from numeric measurements to linguistic reports. Some data cannot be easily quantified, and their accuracy and reliability may be difficult to access.

Sensor measurements have problems related to noise, errors and incompleteness. In addition, we often cannot have a complete view of the world based on data from only one sensor (incompleteness). Associated with sensor data use we have to evaluate its reliability. Reliability attempts to represent how much confidence we have with the data that come from the sensor. All of these aspects contribute to grow the uncertainty in the system. Thus, we need a formal data fusion framework that represents and provides tools to manage all these different problems. None of the frameworks described until now achieves this objective for different types of applications and scenarios.

There are different levels of data fusion. We can have data fusion from one sensor (time series), redundant sensors, redundant variables, variables and systems. We can even fuse different levels of data. We can find different approaches in the literature to treat this problem. Some researchers use statistical analysis like mean, average, median, standard deviation, correlation and variance (the Kalman filter algorithm) [4]. Other researchers use heuristical approaches to manage the uncertainty, such as probabilistic models based on Bayesian networks or uncertainty sets [5][15], possibility models based on fuzzy logic and Dempster-Shafer theory [4][6], mathematical models [7], learning algorithms based on neural networks and evolutionary algorithms [8], and hybrid systems [8]. Which approach to use depends on different aspects, such as the type of data, the requirements of the application, and the grade of reliability desired.

E. Waltez and J. Llinas, cited by [9], have described important features related to the development of data fusion architectures: robustness and reliability, extended coverage in space and time, great data space dimension, reduced ambiguity, and a solution to information explosion. Another important aspect of data fusion is related to system

representation and the data fusion framework. Most of the papers published in this area are related to military applications and image processing. The military application field is mainly represented by the functional model developed by the Joint Director of Laboratories (JDL) from the U.S. Department of Defense. Their functional model is represented by four levels. The first level is related to the identification and description of the objects; the second level represents an interactive process to fuse spatial and temporal entities relationships; the third level is associated with the combination of the activity and capacity of enemy forces to infer their force; and the fourth level is related with all other levels and is responsible for regulation of the fusion process [2]. Although this model has been applied in large-scale projects related to military applications, it seems to be very specific to this field.

The papers related to image processing are basically targeted towards two fields, robot navigation and geographical data. Durant presented a framework to integrate and to model coordination and control of robot systems [10] and Arnoud proposed an architecture of a sensor data fusion system, emphasizing the benefits of providing high level fusion [4]. Clement et al., described a specialist-based knowledge approach [11]. Matsuyama and McKeown worked with hierarchical descriptions of image fusion [12][13]. Grove [14] developed a framework based on semantic nets representation using a fuzzy membership function to determine whether fusion is possible.

A few papers have addressed different domains. Dailey et al., described data fusion applied to transportation [15]. Laskey et al., used Knowledge and Data Fusion in Probabilistic Networks applied to the medical diagnosis domain [16]. The authors describe the use of probabilistic networks to represent and model a medical diagnosis approach.

Although there are different papers in the literature addressing the data fusion problem and the management of incomplete data, there is a lack of a better definition of the different levels of data processing and analysis that need fusion. Some papers address the fusion of signals, others address the uncertainty of high level data fusion using different methods, but none of them have tried to establish a formal framework that includes the different levels of data fusion. Besides this aspect, there is a need for a taxonomy that defines what is low and high level fusion. We present a formal framework based on UML (Unified Modeling Language) and describe a taxonomy that defines the different levels of data fusion. We applied the framework and taxonomy described here to solve the dynamic management of data in a Personal Health Monitoring System (PHMS), but this framework can be applied to any of the class of problems characterized by monitoring the environment using different types of sensors, such as a home security system, a military application in a war, control of robots based on the environment, etc.

In the next sections we present the relationship of the data fusion model with the entire system (section 2), the data fusion taxonomy (section 3) and the Data Fusion

Architecture (DFA) (section 4). Section 5 describes the use of different instances of the Data Fusion Architecture in different domains and applications. Section 6 shows the implementation of the DFA applied to the PHMS application. In section 7 we compare our model to related work and in section 8 we present some conclusions.

## 2. System Architecture

This section shows the relationship of the data fusion with the whole system. We think that to achieve all the requirements to make a system available all the time we need to integrate the network, the service suppliers (data sources) and services consumers (applications). The application development can be divided in several integrated modules; in this paper we will address the data fusion and decision modules.

A network-based system needs all of its components (network, middleware and application) to deal with dynamic changes in the availability of resources and changes in the environment. The system should adapt to the availability of sensors and their corresponding signals, and it should also adapt to changes in the measurements themselves.

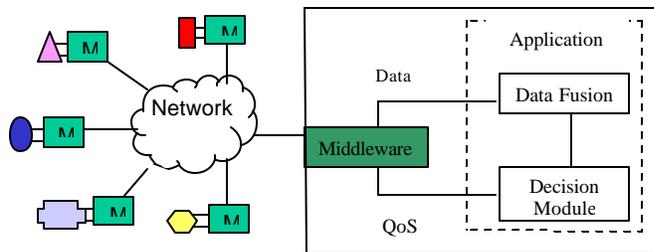


Figure 1: Network, middleware, and application relationship.

Figure 1 shows a block diagram of the system. The sensors (service suppliers) and the application (in this case service consumer) are nodes of the network. The middleware is the software that connects the sensors to the application through the network. The application has two integrated modules, the data fusion module and the decision module. The application sends its QoS requirements to the middleware, and the middleware sends the sensor data to the application. Next we present the data fusion architecture and the decision modules for this system.

## 3. Data Fusion Taxonomy

The taxonomy proposed in this paper divides three main types of data fusion: data oriented, task oriented (variable) and a mixture of data and variable fusion. The basic idea that divides the levels is the difference between *data* and *variable*. Data is a measurement of the environment that is generated by a sensor or other type of source. Variable is determined by an analysis of the data (feature extraction). In general, there can be one or more variables extracted from one type of data. For example, from an image (raw data) an application can determine whether the image contains a person (variable = person-present?), an animal (variable =

animal-present?) or an object (variable = object-present?). These tasks can be an intermediary step or the main goal of the application. The variable determination can be achieved by using a data analysis algorithm or using a probabilistic approach, such as a neural network, that takes the data as input and gives as an output the probability of being an image of a person, an animal or an object. Thus, what determines the three levels is whether the fusion process is made before any data analysis (at the data level), after the data has been analyzed (at the variable level), or is done on a combination of raw data and variables (at the mixture level).

#### 4. The Data Fusion Architecture (DFA)

The framework presented in figures 3, 4 and 5 is new because it explicitly defines the multiple possible levels of data fusion, including different approaches to manage the fusion process. Furthermore, it introduces a new taxonomy for data fusion--classification based on the definitions of data and variable and how to combine and fuse them. The formal representation of a data fusion framework based on UML description is also new.

Sensors exist to measure physical variables, such as a temperature, heart rate, or blood pressure. A physical variable can be measured by several sensors (which we call redundant sensors), or by a single sensor (which we call an individual sensor). Individual sensors are unique in the system, and there is no reason to use multiple sensors of this type. Redundant sensors can co-exist with multiple sensors of the same type, and it is desired to do so to achieve fault tolerance, increase the covering area, or meet other constraints. The system should be able to differentiate these types of sensors and the data that come from them.

In general, before any data fusion can be performed, the signal that comes from the sensor should be pre-processed. The pre-processing can be as simple as an analog to digital conversion or as complex as the use of different digital signal processing approaches. For example, any of the following pre-processing approaches may be useful: error analysis, amplification, filtering, noise treatment, quantization, and multiplexing.

After pre-processing, the data should be fused. We propose a 3-level data fusion framework based on both data and variables. The data fusion can be classified as low level fusion of data, high level variable fusion and a mixture level fusion. When the data fusion is performed before analysis, it is classified as low level. When the data fusion is performed after some data analysis, it is classified as high level variable fusion. There are some situations where we can fuse data and variables. A mixture fusion level class was created to represent this class of fusion. Figure 3 shows the data flow from the sensors, pre-processing unit and the data fusion framework, describing possible scenarios of data fusion represented in UML (Unified Modeling Language).

Figure 3 shows the UML representation of the Data Fusion Architecture. The boxes represent classes and the arrows show the direction of possible data or variable flow.

The numbers around the arrows represent the cardinality of the classes' instances relationship. As an example, we have only one instance of any type of sensor at a time for one instance of a pre-processing class. Text inside brackets represents restrictions, while notes are simple explanations.

The DFA shows that data come from one sensor; they are pre-processed using operations from the pre-processing class and the pre-processed data are sent to the Low Level Data Fusion (LLDF). Each LLDF instance can receive one or more type of data as input and, using any of its operations, fuse them. This class can send its output (fused data) to another instance of the LLDF (multilevel LLDF), and/or to the data analysis, and/or High Level Data Fusion (HLDF), and/or Variable interpretation, and/or Mixture Level Data Fusion (MLDF) modules.

The MLDF receives data from one or more LLDF instances and variables from one or more HLDF instances, and its output (fused data/variable) goes back to the HLDF and/or it goes directly to the variable interpretation module. The data analysis module receives fused data, analyzes it using the appropriate algorithm to the received data and sends the resulting variables to different instances of the HLDF module (one HLDF instance for each type of variable).

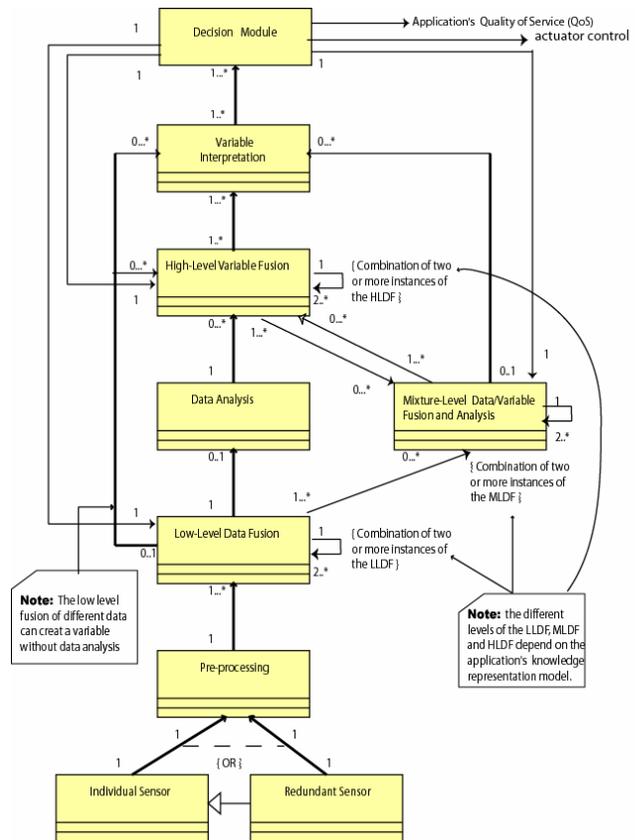


Figure 2: Data Fusion Architecture based on UML.

At the HLDF, each HLDF variable instance performs the redundant variable fusion process and sends its resulting fused variable to another instance of the HLDF, and/or to

the MLDF, and/or to the Variable Interpretation module. The Variable Interpretation (VI) module receives variables from different sources as input. It fuses all the views of the variable to provide as an output a single or multiple views about the sensed environment or data source. Its output goes to the decision system. The decision system takes all this information and decides about modifications that should be made to the environment sensing (application's Quality of Service output), and/or to control any actuator (actuator control), and/or to change algorithms and data/variable flow in the data fusion model (control output goes to the LLDF, MLDF or HLDF).

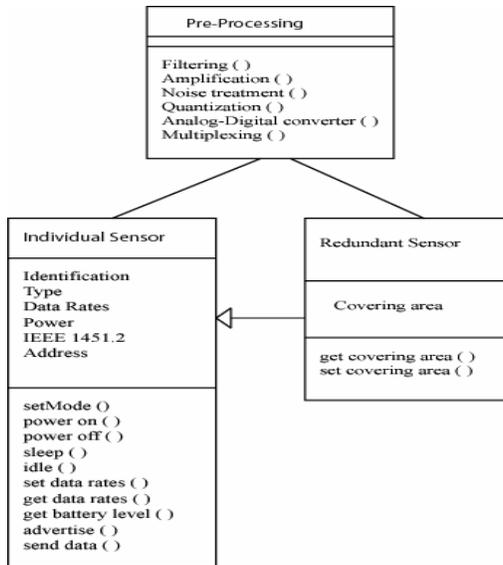


Figure 3: Individual sensor, redundant sensor and pre-processing classes.

Figure 3 shows the classes of the individual and redundant sensors, and the pre-processing classes. The class of individual sensors has attributes such as identification, type of sensor, data rate(s), available energy (lifetime), bandwidth requirement and the IEEE 1451.2 sensor characteristics [19]. The individual sensor class functions are related to the sensor mode (on, off, sleep, idle) as well as data rate control and battery level. The redundant sensor class inherits the individual sensor class attributes and operations and adds the characteristics related to the sensor redundancy, such as covering area and related operations. The pre-processing class includes different functions used to process analog and digital signals.

The low level data fusion class includes different approaches that can be used in different low level data fusion scenarios and is described in Figure 4. Figure 5 shows possible approaches to manage variables in the high-level variable fusion. In general, high level variable fusion uses the same approaches as low level data fusion, but applied to the fusion of variables. The specific variable application fusion management approach includes different approaches based on uncertainty management tools like belief networks, neural networks, genetic algorithms, fuzzy

logic, probability theory and many others. This will depend on the knowledge model applied.

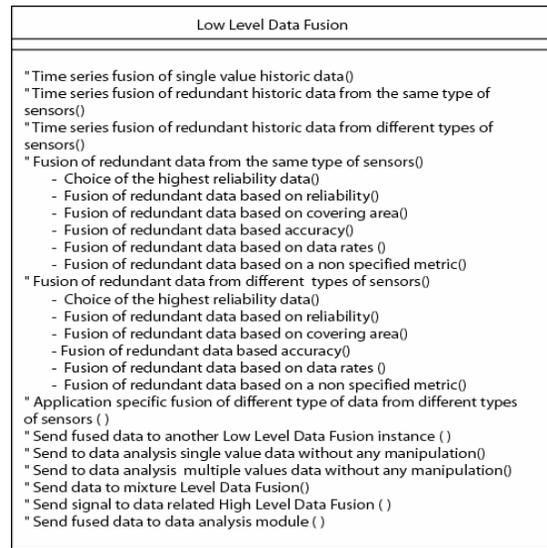


Figure 4: Low Level Data Fusion class.

Figure 6 presents the class of an uncommon but used form of data fusion, the fusion of raw data with variables.

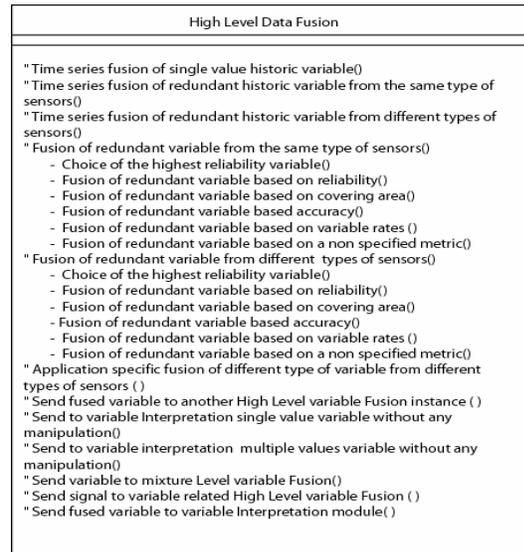


Figure 5: High Level Data Fusion class.

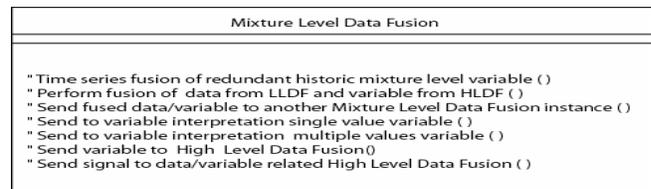


Figure 6: Mixture Level Data Fusion class.

The DFA can be implemented in both centralized and distributed systems. In a centralized system, all raw sensor

data would be sent to the DFA, and the data fusion (low level, high level, and mixture level) would all occur at the same location. In a distributed system, the different fusion modules would be implemented on distributed components. For example, if the sensors are “smart sensors,” the pre-processing and low level data fusion may occur locally on the sensor. The fused data could then be transmitted somewhere else, to be fused with data (raw or fused) from other sensors. The application, network environment, and individual components will dictate where the computation for each of the modules in the DFA occurs.

## 5. Mapping the DFA to Different Domains and Applications

The advantage of the our DF framework is that it can be mapped to many different types of applications in various domains. For example, Figure 7 represents different instances of the DFA. Figure 7a represents the complete system with all the different features of the model. Figure 7b represents an instance of the model using the low and high level data fusion and is capable of creating variables from data without analyzing the data. Figure 7c shows an instance, which does not use the mixture and high level data fusion. Figure 7d shows that this system can only process the data that comes from the sensors and make a decision. Figure 7e shows the use of the low and high level data fusion with analysis of all the data to get variables. Figure 7f allows the creation of variables without data analysis.

These different scenarios show that all the DFA instances have in common a data source, some interpretation of the measured environment (task or variable) and a decision. This is the general architecture of a system, and it can be mapped to different domains and applications, such as military applications, robot navigation, home security systems and health applications.

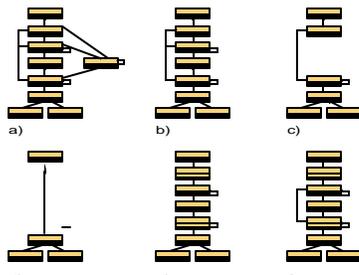


Figure 7: Different instances of the DFA.

In the next subsections we are going to show how to apply the DFA to different domains and applications using the instances described in figure 7.

### 5.1 Military Applications

Imagine the use of smart biological agents sensors deployed in a war area. Using some information

dissemination algorithm applied to sensor networks (e.g., [18]), we can propagate the information of a detected biological agent (data value above a threshold) in the area until it arrives at some point where war plans can be changed or modified (e.g., removing troops from a contaminated area). We can imagine another approach where each soldier carries sensors that advise him, with some grade of confidence, when some biological agent is detected in the environment. Both approaches are represented by the model in Figure 7d. We can also have a more complex system that has some sensors that detect biological agents in the environment and others that detect modifications in the soldier’s body. A neural network can be applied using as input the data from these different sensors and giving as an output the presence or not of an infection resulting from a biological agent. In this case, we are using a low level data fusion approach based on neural networks that directly determines a variable (without performing data analysis). The DFA instance used here is represented in figure 7c.

### 5.2 Robot Navigation

Imagine a robot that, based on different sensors, evaluates the environment and makes decisions about its navigation. This robot can have an accelerometer sensor whose data is analyzed to detect the robot position (variable), and an ultrasound sensor whose data is analyzed to determine (1) the distance from the objects around the robot (distance from object is another variable) and (2) the form of the surface of the objects in the environment (object recognition is another variable). By fusing these variables, the robot can learn about its environment and make decisions to navigate the environment. This system will use part of the DFA represented in Figure 7e.

### 5.3 Home Security System

Imagine a home security application using different sets of sensors like sound sensors, video cameras, ultrasound sensors, temperature sensors, smoke sensors, vibration sensors, and infra-red sensors. From the sound sensor we can determine if a sound is a *human voice*, a *broken window*, an *open door*, or *other types of sounds*; from the camera we can obtain an image or detect the *presence of motion*; from the infrared sensor we can detect *motion*, an *open door*, or an *open window*; from the vibration sensor we can also detect *motion*, an *open door*, a *broken window*, or an *open window*; from the image we can determine whether a *person*, an *animal*, an *object*, an *open door*, or a *broken window* is present in the image. We can see from this example that the variable *person* can come from different types of sensors. All these processes have in common that the data that come from the sensor are processed, redundant data can be fused via low level data fusion, and an algorithm is used to extract the variable for each type of data and fuse the redundant variables in the high level fusion. If we fuse the variable *person* with the variable *broken window* we can

come up with a new variable called *intruder*. This means that we are using a second level of high level data fusion. We can also use a tool that, based on the data from the smoke sensor (without analysis) and the image, can create a new variable called *fire* (mixture level fusion). This entire example will use the model represented in figure 7a.

## 6. DFA Applied to the Personal Health Monitoring System (PHMS)

The heart rate is the result of the electrical activation of the heart, resulting in mechanical contraction of the cardiac muscle. The ventricular contraction results in dynamic changes in blood flow and blood pressure, as well as deformation of the arterial wall (arterial pulse). As a consequence of these physiological aspects, we can measure the heart rate directly by analyzing the cardiac electrical activity or we can measure it indirectly by analyzing the hemodynamic changes. The Electrocardiogram (ECG) is the graphical representation of the electrical activity of the heart. From the ECG analysis we can obtain the heart rate and ECG diagnosis variables. The latter includes, among other aspects, a determination of whether the ECG is normal or abnormal and what abnormality is present (ischemia, infarct, arrhythmia, cardiac chambers enlargement, and other abnormalities).

The heart rate device is a simple device that is based on one ECG lead to determine the heart rate (commonly used in exercise evaluations). The hemodynamic changes can be evaluated through the blood flow and arterial pulse. Different cardiovascular and respiratory exams use the blood flow and the arterial pulse variables to determine the blood oxygen saturation (pulse oximeter), blood pressure (blood pressure device), and blood flow (Doppler). As a consequence, these devices can obtain an indirect measure of the heart rate among other variables.

The blood pressure signal can be fused before any analysis (low level data fusion). After the blood pressure signal is analyzed, we can come up with two different variables: the blood pressure and the heart rate. The heart rate can be measured not only through the analysis of a blood pressure signal, but it can also be measured from an ECG, blood flow or pulse oximeter analysis. As a consequence, we can have the variable Heart Rate from different types of sensors (redundant variable from different types of sensors). If we want to fuse all the heart rate variables, we use the high level variable fusion approach.

Figures 8, 9 and 10 show the application of the DFA applied to the multi parametric PHMS application. Figure 8 shows an example of the use of an individual sensor, pre-processing, low level data fusion and data interpretation classes. The different types and positions of the ECG electrodes require pre-processing that includes the signal amplification; use of a high pass filter (0.5 Hz); use of a low pass filter (25 Hz); use of a notch filter (60Hz); multiplexing; and use of an analog to digital converter. The multiplexing function generates the four types of ECG with different number of leads. The low level ECG fusion

module only forwards the data. This occurs because the management of the redundancy in the ECG data from different leads needs a signal analysis. The ECG analysis algorithm is responsible for recognizing the ECG waveforms and evaluating the grade of normality or abnormality of each wave independently and the sequence of waves in a continuous monitoring. The ECG signal analyzer generates two different variables, the heart rate and the ECG diagnosis.

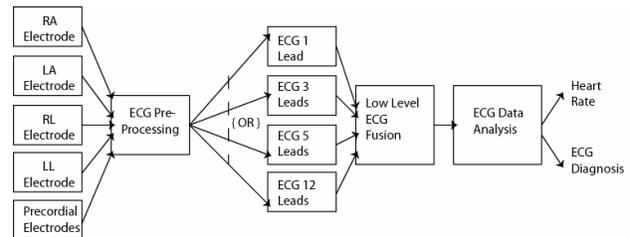


Figure 8: Electrocardiogram pre-processing, low level data fusion, and data analysis.

Figure 9 shows the heart rate variable management. We can measure the variable heart rate from different types of sensors. We can measure it from a Pulse Oximeter sensor (PO), from a Blood Pressure sensor (BP), from an ECG system, as shown in Figure 8 (ECG), from a specific heart rate measuring device (HR) and from a Blood Flow sensor (BF). Each sensor has a different reliability to measure the heart rate. The ECG system has the highest reliability and the Pulse Oximeter has the lowest reliability. Based on power constraints and the system's request for certain reliabilities, the middleware can request from the network a specific set of sensors and present that data to the application. If the application has requested a HR measure with the highest reliability, middleware would present the ECG data (reliability = 1) to the application. The ECG signal would be pre-processed. Then, an ECG analyzer would analyze the ECG data and provide to the Heart Rate high level fusion (HR Fusion) the variable (HR) with a reliability value (R) and the ECG Diagnosis high level fusion (ECG-Diagnosis Fusion) the variable ECG Diagnosis with a reliability value (R). Depending on the system's state, the application can receive and use data from more than one type of sensor.

Redundant variables come from different sensors with different reliabilities and from different locations. Variable is defined as a triple composed by the measured variable, the sensor reliability to measure that specific variable and the sensors placement. For example, the measurement of the heart rate using one Pulse Oximeter sensor placed on the left arm (LA) has a reliability value of 0.7. So, the triple would be represented as (PO-HR value, 0.7, LA). Some of these variables can be objective variables (numeric measures) and others can be subjective (linguistic variables). For example, the measure of the heart rate based on an ECG is an objective measurement. A subjective variable is inferred from a set of interpreted variables. They are subjective assumptions on a specific condition, such as whether the

heart rate is high or low based on the knowledge that the blood pressure is low. Subjective assumptions are defined as a triple composed by the variable V, its reliability measure R (which is very low due to subjective evaluation) and zero, because it is not related to any location (V,R,0).

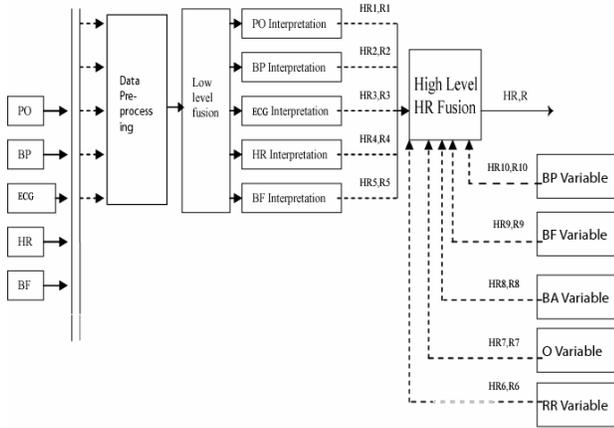


Figure 9: Heart rate variable redundancy.

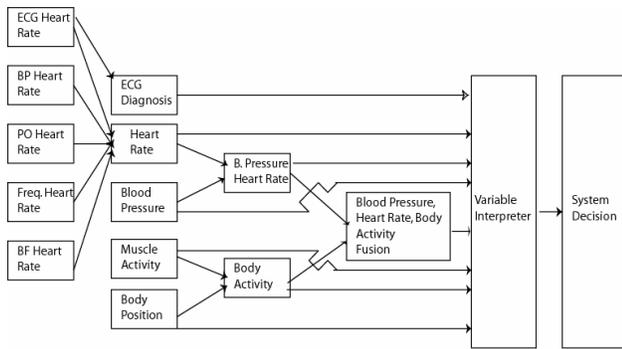


Figure 10: Different level of the High Level Data Fusion in the PHMS application.

At the High Level HR Fusion module, we can fuse redundant disposable heart rate data (HR1 to HR5) and the very low reliability HR predicted data from Blood Pressure, Blood Flow, Body Activity, Oxy gen and Respiratory Rate Interpreted Variables (HR6 to HR10). So, based on the quantity and quality of disposable data, the HR fuser module can perform different procedures.

As shown in Figure 10, the heart rate is combined with the blood pressure and results in the Blood Pressure-Heart Rate fusion variable. At the same time, the Muscle Activity is combined with the Body Position, resulting in the Body Activity variable. The next step is the fusion of the Blood Pressure-Heart Rate variable with the Body Activity variable, resulting in the Blood Pressure-Heart Rate-Body Activity variable. The output of each fusion level goes to the variable interpretation module and after to the decision module. This provides input redundancy to the system decision and guarantees that some decision can be taken in the case of failure of a fusion module.

The current heart rate value has two types of analysis: static and dynamic. The first is related to normal and abnormal interpretation of the variable according to each body's situations. The dynamic evaluation is related to the relative changes (increase or decrease) of the variable. To achieve the static, the interpreter should use a reference table, where heart rate is correlated with the subject's age, to evaluate the expected heart rate basal level. This is very important because the basal HR value of a 1 month old child is near 140 beats per minute, while the basal HR value of a 60 year old man is around 60 to 70 beats per minute. These reference values are useful in static evaluations, but HR Interpretation should consider dynamic situations. As an example, we can consider the case where the body is exercising. In this case, the HR interpretation should consider whether the heart rate is compatible with each level of exercising.

## 7. Discussion

We presented a proposal of a general data fusion architecture described in a formal language of object representations (UML) that tries to represent different scenarios, specifications and features of a general data fusion system. We showed that we can employ our data fusion architecture in diverse scenarios, including different contexts and domains. In relation to the taxonomy employed, although a common taxonomy in data fusion is something difficult to achieve, as discussed in [9], we have covered several different scenarios with our taxonomy.

There are some papers in the fault tolerance related literature that use the terminology virtual sensors to represent what we call in the DFA redundant variables. Thus, we can map the DFA to different models compatible with fault tolerance that employ the same aspect. The different approaches to choose one or more virtual sensors based on events, behavior, or other aspects, is included in the decision module. Besides this, the DFA also allows that every intermediary or final view of the sensing environment arrives at the Variable Interpretation. This aspect guarantees the presence of redundant views and provides a graceful degradation in the view of the environment when different system components fail.

The decision or control module is not a formal part of a data fusion framework. In general it does things not formally related to the data fusion process, such as to control actuators or to modify the sensing. However, if this module controls the data fusion process, it should be included as part of the data fusion model. In the DFA proposed in this paper, the decision module controls the fusion process in different ways, such as changing the data/variable flow, starting new instances of the different DFA modules that depend on different types of data and/or variables and changing algorithms to process or analyze the data. We have explicitly characterized how the fusion process can be controlled by itself.

Most of the papers in the data fusion literature describe different tools to represent knowledge, but very few papers have tried to establish a dynamic model that allows different configurations and different mechanisms to add or delete new features for the model. The DFA provides all these features to apply the same model in different contexts. Furthermore, we employed a well-known language (UML) to represent the model.

The JDL functional model [2] is the most commonly used data fusion model in the literature, although outside the military domain it is not well accepted because it describes features for the data fusion that are difficult to apply in different domains. We can map the JDL model to the DFA in the following way: level 0 corresponds to the pre-processing module; level 1 to the LLDF and data analysis until the point where the variables are generated; level 3 is represented in the DFA by the HLDF and variable interpretation modules; and level 4 is represented by the decision module. All the functionalities provided by the JDL model are provided by the DFA, but the DFA is a more general and dynamic model than the JDL model.

## 8. Conclusions and Future Work

We presented a general data fusion architecture described in a formal language of object representation (UML) that tries to represent different scenarios, specifications and features of a general data fusion system. It also allows a dynamic modification of the system according to different states of the environment or of the system.

The Personal Heart Rate Monitor is from the class of network-based mobile dynamic systems powered by battery, where an application should adapt itself to different configurations of the system (data sources moving in and moving out), different states of the environment, and considering power and bandwidth constraints. The DFA proposed and applied to the PHMS is a good solution to the problem of developing an application framework to manage data from different types of sensors to perform different tasks in a ubiquitous computing environment. We have mapped the data management problem of the PHMS to different domains and applications to show that we can employ our data fusion architecture in diverse scenarios, including different contexts and domains. In this work, we have focused on the application's framework (data fusion and decisions modules). In the future, we plan to develop a dynamic communication approach between the application framework and the middleware. This will allow us to achieve the quality of service of the entire system when there is more than one application running at the same time.

## References:

- [1] Wald L., A European proposal for terms of reference in data fusion. *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXII, Part 7, 651-654, 1998, or Wald L., Some terms of reference in data fusion. *IEEE Transactions on Geosciences and Remote Sensing*, 37, 3, 1190-1193, 1999.
- [2] U.S. Department of defense, data Fusion Subpanel of the Joint Directors of Laboratories, Technical Panel for C3, "Data Fusion lexicon,"1991.
- [3] Wald L., The present achievements of the EARSeL - SIG "Data Fusion". In Proceedings of the EARSeL Symposium, held in Dresden, Germany, June 2000.
- [4] <http://www.science.uva.nl/~aroud/OOAS/Presentation9fus/sld001.htm>
- [5] A. Singhal and C. Brown. *Dynamic Bayes net approach to multimodal sensor fusion*. Proceedings of the SPIE - The International Society for Optical Engineering, 3209:2--10, October, 1997.
- [6] Chen T. M. & Luo, R.C. Multilevel Multiagent Based Team Decision Fusion for Autonomous Tracking System. *Machine Intelligence & Robotic Control*, 1(2), 63-69 (1999).
- [7] M. M. Kokar, J. A. Tomasik and J. Weyman. A Formal Approach to Information Fusion. Proceedings of the Second International Conference on Information Fusion (Fusion'99), Vol.I,pp.133-140,July 1999.
- [8] Myers J.W., Laskey K.B., DeJong, K.A. Learning Bayesian Networks from Incomplete Data using Evolutionary Algorithms. In Proceedings of The Genetic and Evolutionary Computation Conference. 1999. Orlando, Fl.
- [9] <http://www.data-fusion.org/>
- [10] Durrant, H.F. W. Integration. Coordination and Control Multi-sensor Robot Systems. Kluwer Academic Publishers, 1988.
- [11] Clement, V., Giraudon, G., Houzelle, S., Sadakly, F., 1993. Interpretation of Remotely Sensed Images in a Context of Multisensor Fusion Using a Multispecialist Architecture. *IEEE Trans. On Geoscience and remote Sensing*, vol. 31, No 4, pp. 779-791.
- [12] Matsuyama, T., Hwang, V. S.-S., 1990. Sigma: A Knowledge-Based Aerial Image Understanding System, Plenum Press, New York, 277p.
- [13] McKeown, D., Wilson, A., McDermott, J., 1985. Rule Based Interpretation of Aerial Imagery, *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 22(2), pp. 231-243.
- [14] Growe, S. Knowledge Based Interpretation of Multisensor and Multitemporal Remote Sensing Images. <http://www.data-fusion.org/ps/sig/meeting/Spain99ps/growe.pdf>
- [15] Dailey, D.J., Ham, P., Lin, P-J. Its Data Fusion. Technical Report. Washington State Department of Transportation (USA), September, 1996.
- [16] Laskey, K. B. , Mahoney, S. M. Knowledge and Data Fusion in Probabilistic Networks. PhD thesis. [http://ite.gmu.edu/~klaskey/papers/KDFML\\_Laskey\\_Mahoney.pdf](http://ite.gmu.edu/~klaskey/papers/KDFML_Laskey_Mahoney.pdf).
- [17] Carvalho, H.S. Computerized System based on Fuzzy Logic to Evaluate Autonomic Nervous System Based on Multiple Tests. Master Degree Thesis. University of Brasilia, Brazil, 1996.
- [18] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-Based Protocols for Disseminating Information in Wireless Sensor Networks," *Wireless Networks*, Vol. 8, 2002, pp. 169-185.
- [19] K. Lee. "IEEE 1451: A Standard in Support of Smart Transducer Networking."IEEE Instruments and Measurements Technology Conference. May 2000. <http://iee1451.nist.gov/IEEE1451.pdf>

[1] Wald L., A European proposal for terms of reference in data fusion. *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXII, Part 7, 651-654, 1998, or Wald L., Some