A Framework for Multisensory Intelligent Monitoring and Interpretation of Behaviors through Information Fusion

José Carlos Castillo
Instituto de Investigación en Informática de Albacete (l3A), 02071-Albacete, Spain
JoseCarlos.Castillo@uclm.es

Antonio Fernández-Caballero and María Teresa López
Universidad de Castilla-La Mancha, Departamento de Sistemas Informáticos & Instituto de Investigación en Informática de Albacete (l3A), 02071-Albacete, Spain.
Antonio.Fdez@uclm.es
Maria.LBonal@uclm.es

Abstract—Modern intelligent monitoring and interpretation systems manage several kinds of heterogeneous sensor networks and use outstanding segmentation and tracking algorithms. Monitoring has evolved from initial systems based on low-resolution cameras, directly connected to a monitor, up to distributed systems where several sensors cooperate not only to track objects of interest but also to detect suspicious behaviors based on artificial intelligence techniques. In our opinion, frameworks are essential to provide design and implementation patterns for generating a widespread variety of monitoring and interpretation applications, allowing the interaction of different modules and the reuse of code. In this sense, this paper proposes the implementation of a multisensory monitoring and interpretation framework based on the model-view-controller paradigm but extended to distributed intelligent systems.

I. INTRODUCTION

In the last years, monitoring and interpretation systems have experienced a great evolution. The initial systems were based on low-resolution cameras directly connected to a monitor. Nowadays, it is common to work with distributed systems where several sensors cooperate for the purpose of tracking objects of interest and detecting suspicious behaviors. Ambient intelligence (AmI), which refers to electronic environments that are sensitive and responsive to the presence of people, belongs to these kinds of systems. In our opinion, the definition of a framework is essential to provide design and implementation patterns able to generate the widespread variety of monitoring and interpretation applications. Such a framework enables the interaction of the different modules of an AmI system. Also, it is capable of reusing the code from one application to another [8].

Obviously, any new approach to intelligent monitoring and interpretation should be adaptive when considering the incorporation of an ever growing and wider range of sensors. Therefore, modern monitoring frameworks have to manage and merge several kinds of heterogeneous sensor networks and use the most outstanding segmentation and tracking algorithms. This is why fusion techniques become a key issue for complex behavior interpretation and situation analysis. Information fusion of data from several sources, in general, provides higher accuracy and robustness against failures. It also helps reducing noise and uncertainty while obtaining better resolution of the measures.

Also, AmI can be designed as a distributed layered architecture enabling ubiquitous communication and an advanced human-machine communication protocol. Thus, synergies between the user and the environment rise, enhancing training of professional skills, and improving security and safety in the human environment [18]. For instance, the PRISMA surveillance system [11] combines audio sensors and security cameras for improving security and safety in public transport networks. The system consists of a network of distributed processing systems (CCTV, IP cameras, audio sensors) and a central node that collects data from remote systems and implements the user interface.

The most popular trends in monitoring and interpretation systems are people segmentation and tracking as well as human activity recognition (e.g. [7], [19], [4], [3]). Current monitoring systems are composed of a number of sensors covering wide areas, grouped in processing nodes that allow high scalability and robustness. The nodes perform real-time monitoring to warn the human operator about pre-defined events. Some common requirements fulfilled are fusion of different sensorial information [2], network restrictions awareness [17], quality of communication among nodes [17], and automatic learning [1].

Lately, many proposals are centered in the combination of several cameras for human tracking and activity analysis. For instance, a system is implemented for football players classification and tracking [20]. The system is composed of eight cameras, each one provided with a processing unit. In the context of home monitoring, the “EasyLiving” system is proposed [10]. In this case, a multicamera tracking system is used for behavior recognition in homes. Finally, there are also approaches for outdoor monitoring. One example is the use of heterogeneous sensor networks for multimodal tracking of vehicles within a human environment [9].

The generation of a new multisensory monitoring and interpretation framework is proposed in this work. The framework is based on the model-view-controller paradigm, where some features are extended and adapted to the requirements of a distributed intelligent system [6], [15]. The proposed hybrid distributed system is composed of a central node and a set of remote nodes. The remote nodes are in charge of data acquisition and lower level processing (e.g. segmentation and tracking) whereas the central node collects information and performs higher level processing tasks (e.g. activity detection).
II. THE MVC PARADIGM

Model–view–controller (MVC) [16] is a paradigm widely employed in software engineering. In this paradigm the logical domain is isolated from the user interface to allow independent development, testing and management. Thus, applications are divided into three parts. The model is in charge of managing the application data (coming from databases, text files, XML files, computer registers, etc.), initializing application objects, providing log information and primitives to update the system state (through its objects). Besides, the model notifies information changes to the other system components, view and controller. Additionally, the view provides the representation of the model, adapted to the user needs, even generating several views of the same model (multimodality concept). As the view does not contain the application logic, it must be kept as simple as possible. Finally, the controller receives the inputs, invoking model primitives to update its objects. The controller receives the user’s inputs (listening to the view changes) that trigger actions in the model and view.

A. Extending the MVC paradigm

The proposed work extends the traditional MVC architecture to provide a higher flexibility. This is necessary not only to fulfill the needs of monitoring and interpretation systems derived from the framework but also to incorporate the existing algorithms without great design changes. For these purposes, the business logic is detached from the model, generating a new execution unit managed by the controller. The new unit is named algorithm, and its functionality is given by the different modules composing the framework levels. This way, it is possible to easily incorporate newer or existing functionalities to the framework, not only by the framework developers but also by other users interested in testing their proposals. Each level is implemented as MVC extended modules (see Fig 1), operating from the lower level - data acquisition or nodes communications - to higher levels such as tracking, classification or activity detection. Here, the model functionality is reduced to application data storage and management, providing primitives to access the data.

In the traditional MVC paradigm all elements are included within the three original modules. But, the proposed extension considers the framework as a combination of MVC extended modules. Each module is devoted to one level of the proposed monitoring and interpretation framework. Nevertheless, the framework modules are integrated through the addition of a new module that holds the data model and the backbone to integrate the rest of modules. This module implements the common model (see Fig. 3, seen as the spinal cord of the framework, since it contains the data generated by each level, offering it to the upper levels through a set of interfaces. The common model reduces the dependencies among modules and simplifies the access to the data through a set of well-defined input and output methods. Later on, the main features of the common model are defined (see Table I). Also, a common view is defined to contain each module’s interface. Through the view, each module provides an interface to show its results and to tune parameters on-line. Finally, the module also implements a controller to monitor the execution of all framework modules.

III. THE PROPOSED MVC-BASED FRAMEWORK

A. Execution model

Prior to detailing each framework level, it is necessary to describe the execution model. The execution model follows a hybrid distributed scheme where remote nodes perform low-level processing and a central node is in charge of collecting and fusing information as well as performing high-level processing. In Fig. 2 a schematic representation of the framework modules is shown. The remote modules incorporate the MVC extended structure. The common model, the global controller and the common view are also held in the central node with the MVC extended structure.

B. Layers of the framework

The framework levels composed of MVC extended modules are provided in Fig. 3. In first place, the common model stands out. As aforesaid, this block is accessible from every level. The functionality of each level is described next. Notice that no algorithm is provided to keep the framework as generic as possible. Of course, the proposed levels are just a guideline to create the framework, but it is possible to include new levels according to the application requirements.

The Acquisition level directly interacts with the digital analog devices through measuring from the physical world. The measures are data from the sensors as well as data from other information sources (disk, database, and so on). The Acquisition level also performs information preprocessing. The Sensor Fusion level merges the sensor data to improve...
the remote nodes (see Fig. 2). The rest of the layers are implemented in the central node together with the common model, the central controller and the central view.

In a multisensory monitoring and interpretation framework, where several sensors monitor a common scenario, the events generated from different sources do usually not match. This is why the Event Fusion level is necessary to unify the information arriving from the different sensory data generated in the previous level. The final level of the framework, Activity Detection, is in charge of the analysis and detection of activities already associated to temporal features. After Event Fusion, the current level has a better knowledge of what is happening in the scenario according to the detected events. Hence, the activities detected at this level can be translated into actions along the scenario, providing a higher level of abstraction.

IV. DEFINITION OF THE COMMON MODEL

A new layer, known as the common model, is considered to simplify the information exchange among the framework layers. The layer gathers all the information from the different levels while providing primitives to access the information. The common model is a variation of the traditional MVC model. Indeed, the common model is only in charge of holding the common information to be accessed by each execution module. For this purpose, the primitives that allow managing the data are provided. To properly define the common model, we will start with the layers that compose the framework. Since the input and output parameters are known, it is possible to estimate which of them belong to the common model. Table I introduces some of the most important parameters and functionalities that the common model provides at each framework level.

V. CONCLUSIONS AND FUTURE WORK

This paper has introduced a proposal for a monitoring and interpretation framework which consists of a distributed layered architecture enabling ubiquitous communication and an advanced human-machine communication protocol. The framework is based on the model-view-controller paradigm, although it is improved to deal with the requirements of a distributed multisensory monitoring framework. Moreover, the proposal presents an execution model based on a hybrid distributed scheme to allow ubiquitous computation and communication between a central node and remote nodes.

The work is currently focused on the generation of a meta-model able to manage the modules’ hierarchy needed to implement all framework layers. On the other hand, the common model is also being defined in order to populate a standard set of inputs and outputs that all modules have to provide. This is a necessary constraint to allow an easy integration among modules in different levels. Finally, monitoring algorithms will be explored to test and enhance the framework’s functionality. For this purpose, it will be necessary to study their operation as well as the parameters required as input and output. Indeed, the common model
input and output primitives have to be designed flexible enough to fit them.

VI. ACKNOWLEDGMENTS

This work is partially supported by the Spanish Ministerio de Ciencia e Innovación under project TIN2010-20845-C03-01, the Spanish Ministerio de Industria, Turismo y Comercio under project TSI-020100-2010-261 and by the Spanish Junta de Comunidades de Castilla-La Mancha under projects PI2109-0069-0994 and PEI109-0054-9581.

REFERENCES