AVITRACK

Task 5.1
Framework Prototype

1. SUMMARY

The document is the summary of the deliverable 5.1 related to the Work Package 5 “Prototyping” of the AST3-CT-3002-502818 project AVITRACK.

Work Package 5 is dedicated to the framework prototype, from the design (task 5.1) to the integration of all software modules.

Main results reached at this stage are as followed:

- The definition of AVITRACK distributed architecture was done,
- Smartsoft (CORBA based communication framework) is chosen as the communication framework between all software modules.
- Development will be done in C++ on Linux.

2. GENERAL OBJECTIVES

In the frame of the project, it was decided that using a software framework to build the Avitrack was **mandatory** because of:

- the complexity of the task to be achieved,
- the unknown needs of the user,
- and the integration of Avitrack in a complex information areas (Airport) are constraints that can be completed only using a **robust** and **opened** software framework.

The goals of the Avitrack project is not only to study the feasibility of activity recognition on an apron area but also to build a prototype that shall be later used for future Avitrack integration of new component.

3. REQUIREMENTS

Derived Requirements for the software framework issued by the analysis performed during the phase are:

Open system

It is planned that the prototype will be inserted in a complex environment where several information systems co-exist (Air Traffic Control systems, Airport information systems, companies and handlers networks…). As such, the framework is designed to allow new connections to those systems to retrieve new information sources (ground radar image…) or to provide services to the Avitrack customers without re-designing the framework.

Up to now, the user-needs are not identified (one work package is assigned to this task), so the software framework should response to needs that will be defined in the future. Such constraints could be achieved by using an open-system.

The design should use the client/server paradigm: thus adding new components or new source of information should not impact the software design. All software modules/components shall have information about all necessary data sources locations.

Object-oriented components - Modularity

An object-oriented design was adopted leading to a clear identification of the different components of the Avitrack architecture.

Benefits resulting of a modularity design are:

- A structured platform/framework is generally easy to work, to develop and to maintain.
- Independency between components: easy development, debugging and maintenance. Each component can/shall be maintained by an entity without necessity of changing third party modules and without interfering with "surrounding" modules.
- The inclusion and replacement of new/old modules, for instance sensor interface modules, shall be easy. The AVITRACK project will evolve to the inclusion of new cameras, sensors and third party hardware sensors. Modularity, in the aspect of API (communication pattern/protocol) definition is very important to achieve high modularity level.

Scalability – Network distribution

Scalability was considered as a key issue for the future AVITRACK framework. The expansion of the number of cameras, additional sensors and related hardware is an assured issue regarding the present (prototype) scope: in order to be used in larger airports, framework requires a high level of scalability.

Design constraints set up to cope with this requirement are as follow:

- AVITRACK framework shall be able to be network distributed,
- The SW components distribution on the HW architecture shall be realized according to their definition, specification and principally according to the amount of "work" they perform, either dynamically or by configuration file (statically). Prioritisation of processes, as SW distribution to unused process, has also been taken into account.

Run time performances

AVITRACK is a real-time platform that shall be able to track objects/persons. The performance is also a key issue for final user to not loose track of a given item.

Main topics were addressed during the project, like:

- choice of SW programming languages,
- budget analysis, where data coming from 8 cameras were taken as assumption for storage and data transfer

Security

The information inside AVITRACK platform shall be protected (especially the access to the recorded video). Two solutions were addressed, but will not be implemented into the framework because of lowering the performance.

4. SOFTWARE TOOLS FOR THE FRAMEWORK

Three kinds of solutions were evaluated during this phase to build the software framework. Each of them corresponds to a different level of abstraction:

- Communication libraries
- Object distributing capabilities
- Component based approach.

In this chapter, an evaluation was made on:

- existing software framework based on the component approach,
- communication tools that can be used to build our own software framework (CORBA or libraries).

A pre-selection phase shows that the most suitable solutions were: ACE/TAO middleware, gSOAP, and Orocos::Smartsoft.

Orocos::Smartsoft implements all project needed, is easy to use and will integrate Avitrack changes or extra development after being tested. **Therefore, Orocos::Smartsoft is chosen by the partners**.

5. ARCHITECTURE DESIGN

5.1. COMPONENT DECOMPOSITION

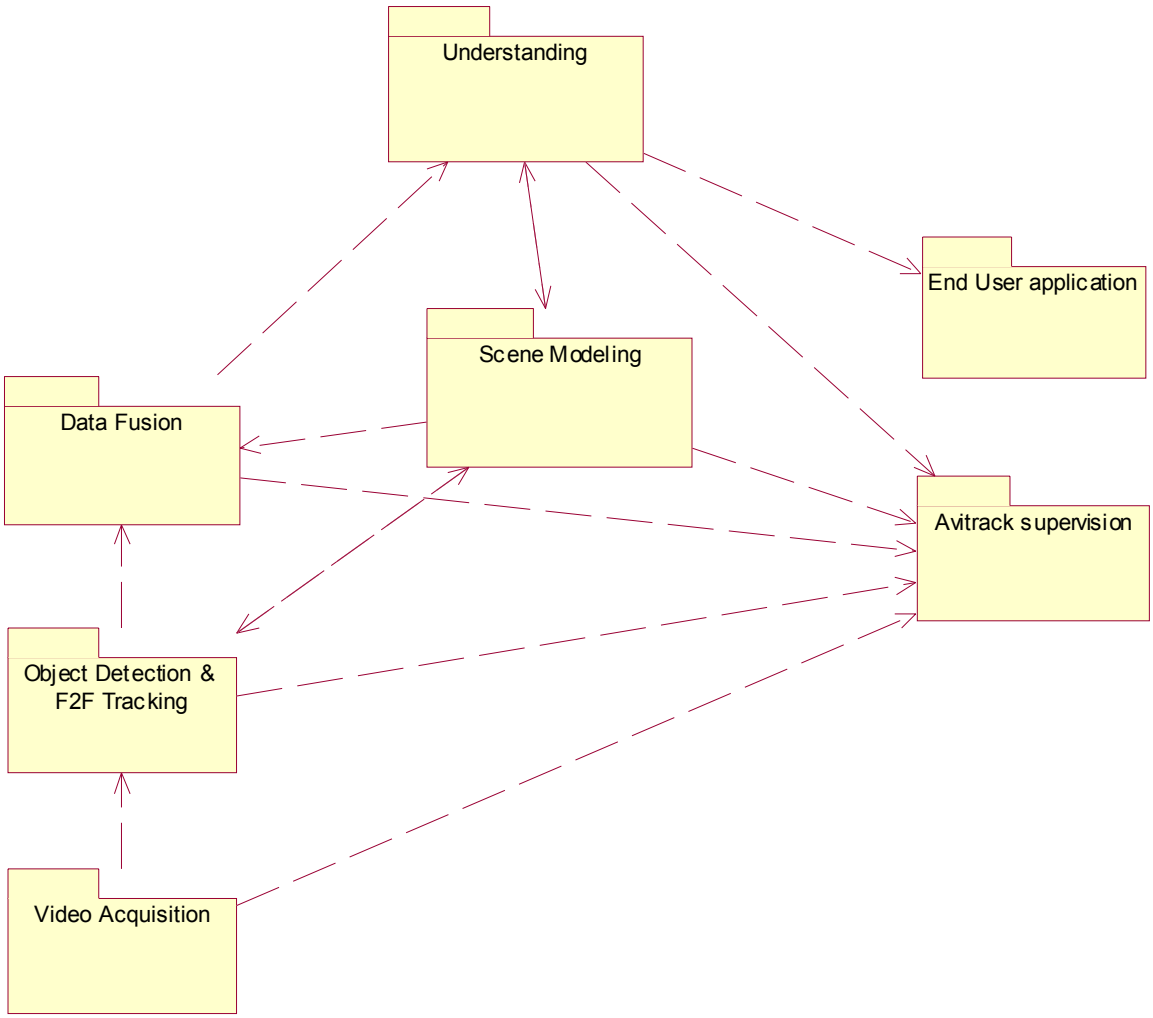

Figure 1 Module decomposition

5.2. COMMUNICATION PATTERNS

The communications patterns for the Avitrack prototype will be those of the Orocos::Smartsoft approach. It means that interaction between components will occur **only** using these messages.

Using such communication patterns allows:

- To increase independency between components,
- To simulate components (development phase),
- To easily replace a component by another (old – new)
- To easily add new component (sensors, …).

Each data exchange between components should be carefully studied to extract:

- The possible clients for that information,
- The possible sources for that information,
- The applicable communication pattern.

This study shall include new sensors (cameras, ground radar information) or extra information sources (airport, air traffic control, companies, handlers…).

Figure 2 Module integration shows how each component should be integrated in the software framework.

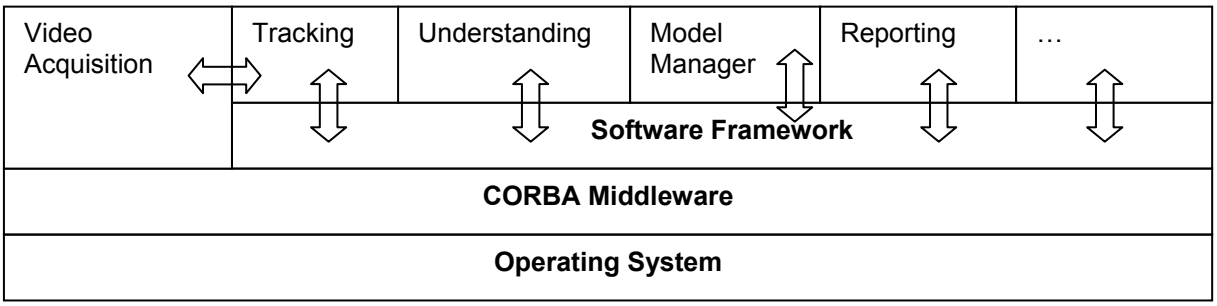

Figure 2 Module integration

6. HARDWARE ARCHITECTURE DESIGN

Avitrack shall process video images from 8 cameras at the same time; thus, a huge computational power is needed.

A new deliverable: “AVI-D5.1-b-Hardware-design” will be created to address all hardware related issues that cannot be fixed now. It will contain the precise description of the architecture and a detail list of all hardware needed to build the framework prototype.