

AVITRACK



Contract n° AST3-CT-2003-502818

D.5.3 Models Manager

Abstract: This document presents the architecture of the Models Manager component, included in the AVITRACK prototype. Its function is to give an access to models for input data for all other software components.

1. THE OBJECTIVE OF "MODELS MANAGER"

Avitrack project aims to recognize and understand activities on the apron. To reach these tasks the system needs some models of the scene, objects involved.

The objective of "models Manager" is to obtain an access point, visualization and an end-user representation of all models used in the Avitrack prototype.

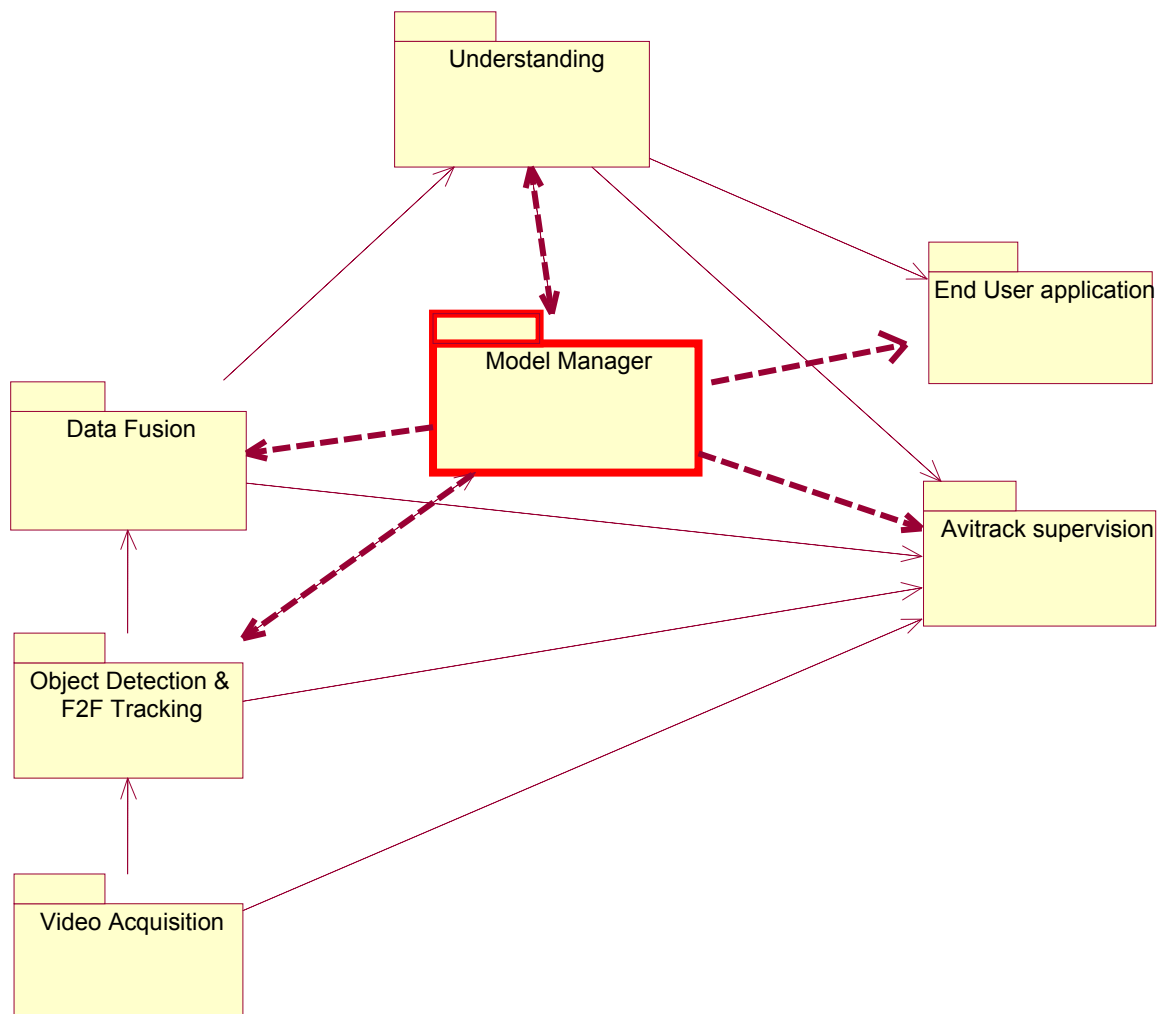
The "Model management Module" gives access to the model formulation for input data taking account of models of the scene, objects involved (vehicles, plane, jet bridge, stairs), ...

2. THE ARCHITECTURE OF "MODELS MANAGER"

2.1. MODEL DISTRIBUTION

Avitrack prototype is operating in a distributed environment. Therefore, the Models Manager module will be the only access point to all models used in the prototype.

Model distribution will be done using the framework tools describe in deliverable D5.1 "Framework prototype".



2.2. COMMUNICATION MODEL

The communication used between the Model Manager and a client should address specific needs. These needs are independent of the type of model.

The needs expressed are:

1. **List Query:** A client can retrieve the list of models available for one category.
2. **Model Query:** A client can query one model from its identifier and category.
3. **Model Update:** A client can update a model by sending its new data.
4. **New Model available:** The server informs clients that a new version of a specific model is available.

2.3. COMMUNICATION OBJECT

The communication will use a dedicated object to handle ASCII data. This object allows components to stream data into it, to send it as a Framework communication object using any of the available communication patterns.

2.4. NEEDED SERVICES

For one category of model, the model manager will create:

- A service to answer to queries.
- A service to receive model update.
- A push service to inform clients that a new model is available.

All the services are created automatically for all categories.

2.5. MULTIPLE CATEGORIES SERVICING

The Model Manager module serves several types of models. Thus, the server is designed to dynamically create all services needed for one category from a configuration file. This file used an XML based format.

3. RESULTS

The Model Manager can be seen as a repository, centralising all models and distributing them to process units.

The Model Manager module is a dynamic model.