13.1 Introduction

Computer application basic inputs, such as keyboard strokes or pointing devices, supply only limited information about the surrounding environment. The necessity of context information grows as applications need to adapt to the environment in which they are used. This adaptation increases the application’s performance and makes sure that the results are well adapted to the specific circumstances. The main objective of context-aware computing is the development of applications that, without being limited by usual input devices, acquire and use context information to better adapt to the circumstances in which interactions take place.

Since context can be acquired by a wide range of input devices (i.e., domain dependent sensors), context systems were created to provide a simple and unique source of information for applications. Context systems can then be understood as extensions of the basic input that an application can receive [4].

This chapter describes the CASCOM approach to context acquisition and management — GCMAS (General Context Management and Acquisition System). The proposal was developed under the assumption that specific information is considered context if it complies with the following definition:

“Context is all the information related to persons, objects, locations and applications, participating or being referred in a specific interaction, which is not strictly necessary for the interaction to be accomplished, although the use of this information allows improving the quality of the interaction and often the system’s performance.”

The presented definition integrates aspects from the definitions put forth by Anagnostopoulos et al. [1], and by Dey and Abowd [4], which are probably the most accepted ones in the scientific community. It emphasizes the central role of interactions, and sets a clear distinction between information that is essential to the interaction and context information which, although not being essential, may
be used to improve it. This distinction is humbly suggested by the authors of this chapter with the goal of providing some guidance about the difference between context-aware computing and computing in general. According to this view of context information, while non-context information (i.e., information that is strictly necessary for the task at hand) should be explicitly provided to the system when requesting it to perform some task, context information (i.e., information not strictly necessary) should be acquired by the system that receives the request, even if it has to ask it back to the requester.

The following example may shed some light on this issue. Imagine someone, say Tom, that wants to use the CASCOM Agent System to locate a Healthcare Centre. Tom might send a request to the CASCOM Agent System saying "Find me a Healthcare Centre". This is the information explicitly sent in the message initiating the interaction. However, being a context-aware system, the CASCOM Agent System knows that Tom’s location is important for discovering a Healthcare Centre more appropriate to Tom’s situation. Therefore, it asks Tom’s location to GCMAS, the CASCOM Context System. Having Tom’s location, the CASCOM Agent System will be capable of discovering the Healthcare Centre closest to Tom. In this example, Tom’s location is context information hence acquiring it is the responsibility of the CASCOM Agent System.

Other context definitions may be found in the context-awareness state of the art chapter (Chapter 5), in particular in Section 5.2.

GCMAS is responsible for acquiring, monitoring, representing and storing context information. The system is organized in two layers - an infrastructure layer and an application layer. The infrastructure layer incorporates mechanisms that allow applications to provide context information, request context information and subscribe information about selected context events. Application dependent context processing mechanisms are included in the application layer, which provides mechanisms for context modeling, aggregation, and reasoning adapted to the type of application that accesses the system.

The chapter begins with the system requirements. Following, it presents context representation decisions, namely the content and structure of the ontologies used to model context in GCMAS. Next, it presents the description of GCMAS architecture explaining each of its components and functionalities. Finally, in the last section, it presents a discussion of the context system, briefly describing an example in which GCMAS is used in one of the CASCOM medical emergency scenarios, and presenting results and conclusions.

13.2 System Requirements

This section presents a set of requirements assuming, without losing generality, that context information is acquired by sensors, which is in fact the most accepted choice. The described requirements are classified in two types: functional and non-functional. The discussed requirements were identified through a review of the