

Technical Report - DIAB-18-07-1

Glossary of terms for Context-Aware Systems

Cristina Roda¹, Uwe Zdun², Elena Navarro¹, Víctor López-Jaquero¹

¹LoUISE Research Group, University of Castilla-La Mancha, Spain

{Cristina.Roda, Elena.Navarro}@uclm.es, victor@dsi.uclm.es

²Software Architecture Group, University of Vienna, Austria

uwe.zdun@univie.ac.at

In this technical report, it is presented a glossary of terms collected throughout the entire process of this mapping study. The main reason for preparing it was that, while analysing the different papers, we realized that different terms were used indistinctly for describing the same thing. This compilation of concepts will be useful for researchers interested in context-aware systems and their software architecture infrastructure. Next, these terms are presented, related to the dimensions of context (Section A. 1), the software architecture concepts (Section A. 2), and the research topics of the selected papers, among others (Section A. 3). Notice that all terms are ordered alphabetically.

A. 1 Dimensions of Context

Some terms were found associated to the dimensions of context and they are represented in Table 1. Notice that the first column corresponds to those terms most commonly used by the research community.

Table 1. Glossary of terms related to the dimensions of context

<i>Term</i>	<i>Similar terms</i>	<i>Context dimension</i>	<i>Description</i>
Content	Information	Platform	Information formatted and presented in the UI, e.g. pictures, tables or text.
Gender	Sex	User	Physical/social condition of the user, such as female or male.
Historical data	Context history	Other	Information from the past, such as the interaction history of the user, usually used to know the evolution of the user and hence making changes and adaptations in the system that improve the user experience.
Light	Luminance Bright Brightness Illumination	Environment	Brightness that comes from one or several points, e.g. from the sun, a lamp or a fire.
Noise	Sound	Environment	Any type of sound that comes from the surrounding environment.
Physiological data	Physiological context Biological signals Physiological status Physiological signals	User	Information associated to the biological signals of the user, e.g. heart rate, skin conductance, ECG, etc.
Preference	Need Requirement Interest Demand Convenience	User	Anything desired by the user of the system, e.g. a light colour for the UI.
Presentation	Format Layout	Platform	The way the components or widgets are located and formatted in the UI.
Screen	Display Monitor	Platform	Surface of some devices, e.g. smartphones or computers, on which pictures and other type of content are shown.
Weather	Climate	Environment	Conditions in the air, e.g. rain, fog, wind, temperature or humidity.

A. 2 Software Architecture Concepts

Table 2 presents a glossary focused on describing the different software architecture concepts found for developing context-aware systems.

Table 2. Glossary of software architecture concepts

<i>Software architecture concept</i>	<i>Description</i>
Aspect-Oriented Architecture (AOA)	An aspect represents a specific concern (safety, coordination, distribution, etc.) that crosscuts the software architecture, i.e. those concerns that do not crosscut the architecture will not be an aspect [38].
Blackboard architecture	The information of the computation is centrally located and operated on by independent computations which interact only through the shared data [21].
Client-Server architecture	Particular case of distributed architecture in which a server is a process that provides services to other processes, i.e. the clients. Usually the server does not know in advance the clients that will access it at runtime, but clients know the identity of the server and can access it by remote procedure call (adapted from [21]).
Component-based architecture	The system is represented as a set of software components, their connections and their behavioural interactions [4].
Distributed architecture	Distributed systems have developed some common organizations for multi-process systems. They can be characterized by their topological features, such as ring and star organizations, or by the types of inter-process protocols that are used for communication (adapted from [21]).
Event-triggered architecture	A component broadcasts one or more events, while other components register an interest in an event by associating a procedure with such event. When the event is announced, the system invokes all procedures that have been registered for that event (adapted from [21]).
Micro-architecture	Micro-architectures are building blocks for designing applications. They represent a higher level of abstraction than individual patterns, and are expressed by a combination of patterns to solve a problem (adapted from [3]).
Middleware architecture	A middleware decouples the individual software components of the infrastructure from one another, presenting a uniform level of abstraction (adapted from [26]).
Model-Driven Architecture (MDA)	A conceptual framework that separates business decisions from platform decisions to provide flexibility when building and evolving the system (adapted from [6]).
Multi-Agent System (MAS)	For modelling real-world and social systems where problems are solved in a concurrent and cooperative way without the need of reaching optimal solutions. Its architecture is made of agents which are reactive, proactive and exhibit an intelligent and autonomous behaviour (adapted from [47]).
Multi-layered architecture	An architecture organized hierarchically where each layer provides service to the layer above it and serves as a client to the layer below. The connectors are defined by the protocols that determine how the layers interact (adapted from [21]).
Peer-to-Peer (P002P) architecture	This architecture provides a network characterized by self-organization, symmetric communication (all nodes are both servers and clients) and distributed control (no centralized server) that automatically rearranges itself to joining and leaving nodes (adapted from [42]).
Pipe-and-filter architecture	Each component (filter) has a set of inputs and outputs and reads streams of data on its inputs and produces streams of data on its outputs, delivering a complete instance of the result in a standard order. There are connectors that transmit outputs of one filter to inputs of another so they are called pipes (adapted from [21]).

Plug-and-Play (PnP) architecture	Architecture made of pluggable components so they can be reused or replaced with others easy and seamlessly.
Publish/Subscribe architecture	Information providers publish events to the system, and information consumers subscribe to events of interest within the system. This architecture ensures the timely notification of events to the interested subscribers (adapted from [62]).
Service-Oriented Architecture (SOA)	Architecture composed of collections of services, connected via well-specified contracts, which make calls on operations defined through their service interfaces (adapted from [6]).

A. 3 Research Topics and Other Terms

Some terms were found associated to the research topics of the selected papers and they are represented in Table 3.

Table 3. Glossary of terms related to research topics, among others

<i>Term</i>	<i>Similar terms</i>	<i>Description</i>
Context-aware system	Situation-aware system Context-sensitive system	System that is able to adapt its behaviour based on the changes of the surrounding context.
Context delivery	Context dissemination Context sharing Context provisioning Context transferring Context distribution Context provider Context provision	When an event occurs and matches any trigger previously specified, then some context information is provided to an application or service that requires it (adapted from [18]).
Context discovery	Context detection	Acquisition and inference over low-level context information to translate it into a more useful format for the application, i.e. high-level context information (adapted from [10]).
Context gathering	Context acquisition Context extraction Context recognition Context collecting Context monitoring Context sensing Context storing Context assimilation Context retrieval Context consumption Context query Context access	First step of context discovery process in charge of collecting context data or subscribing to event notifications from different sources and sensors (low-level context information), usually demanded by services and applications (adapted from [43]). These sources have to model the context according to the format previously established and communicate with the system using the predefined set of interfaces and protocols (adapted from [51]).
Context inference	Context reasoning Context interpretation Context deduction Context integration	Some context information cannot be obtained directly, but can only be derived to transform it into high-level context information. For example, the information whether or not a person is travelling and with which means of transport cannot be sensed directly and it has to be reasoned from other context information, such as Wi-Fi

	Context aggregation Context composition	access points or calendar information. It represents the final step in the context discovery process (adapted from [53]).
Context management	Context processing Context handling Context handler	Resolving possible conflicts between contradictory sources of context information. It is in charge of processing the context information gathered by the context discovery component (adapted from [10]).
Context modelling	Context representation	Representing context information, by describing context attributes, their relationships and their values (adapted from [53]).
Service	-	A self-described, loosely coupled, modularized, self-contained and platform-independent software component that can be discovered and requested to implement a particular functionality (adapted from [11]).
Service composition	Service reconstruction Service integration Service customization Service reconfiguration Service recomposition	Combination of some services together into a composed service to fulfil a complex business requirement (adapted from [11]).
Service delivery	Service provisioning Service provision Service suggestion Service recommendation Service provider Service presentation Service sharing	Provision of some functionality in the form of services, by answering simple requests or executing sophisticated processes. Services must be formally described to bring their descriptions in correspondence with existing ontologies (adapted from [56]).
Service discovery	Service matchmaking Service access Service selection Service search Service consumption	Resolving consumer requests in terms of advertised services which involves the selection of appropriate services and their prioritization. Service discovery can be done manually, automatically or by a combination (adapted from [56]).
Service management	-	Organizing services by domain or location in order to be later discovered and consumed by applications (adapted from [43]).