Short introduction to basic “CAFCR” model

Abstract
The basic “CAFCR” reference model is described, which is used to describe a system in relation to its context. The main stakeholder in the context is the customer. The question “Who is the customer?” is addressed.

Distribution
This article or presentation is written as part of the Gaudi project. The Gaudi project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

All Gaudi documents are available at:
http://www.gaudisite.nl/

version: 0.4 status: draft September 9, 2018

1 Introduction

A simple reference model is used to enable the understanding of the inside of a system in relation to its context.

An early tutorial[2] of this model used the concatenation of the first letters of the views in this model to form the acronym “CAFCR” (Customer Objectives, Application, Functional, Conceptual, Realization). This acronym is used so often within the company, that changing the acronym has become impossible. We keep the name constant, despite the fact that better names for some of the views have been proposed. The weakest name of the views is Functional, because this view also contains the so-called non functional requirements. A better name for this view is the Black-Box view, expressing the fact that the system is described from outside, without assumptions about the internals.

The model has been used effectively in a wide variety of applications, ranging from motor way management systems to component models for audio/video streaming. The model is not a silver bullet and should be applied only if it helps the design team.

2 The CAFCR model

A useful top level decomposition of an architecture is provided by the so-called “CAFCR” model, as shown in figure [1]. The Customer Objectives view and the Application view provide the why from the customer. The Functional view describes the what of the product, which includes (despite the name) also the non-functional requirements. The how of the product is described in the Conceptual and Realization view, where the conceptual view is changing less in time than the fast changing realization (Moore’s law!).

![CAFCR model diagram]

Figure 1: The “CAFCR” model

The job of the architect is to integrate these views in a consistent and balanced way. Architects do this job by frequent viewpoint hopping: looking at the problem...
from many different viewpoints, sampling the problem and solution space in order to build up an understanding of the business. Top-down (objective driven, based on intention and context understanding) in combination with bottom-up (constraint aware, identifying opportunities, know how based), see figure [2].

In other words the views must be used concurrently, not top down like the waterfall model. However at the end a consistent story-line must be available, where the justification and the needs are expressed at the customer side, while the technical solution side enables and support the customer side.

![Diagram](image)

Figure 2: Five viewpoints for an architecture. The task of the architect is to integrate all these viewpoints, in order to get a valuable, usable and feasible product.

The model will be used to provide a next level of reference models and submethods. Although the 5 views are presented here as sharp disjunct views, many subsequent models and methods don’t fit entirely in one single view. This in itself not a problem, the model is a means to build up understanding, it is not a goal in itself.

### 3 Who is the customer?

The term *customer* is easily used, but it is far from trivial to determine the customer. The position in the value chain shows that multiple customers are involved. In figure [3] the multiple customers are addressed by applying the CAFCR model recursively.

The customer is a gross generalization. Marketing managers make a classification of customers by means of a market segmentation. It is recommended to
Figure 3: CAFCR can be applied recursively

start building up insight by making specific choices for the customer, for example by selecting specific market segments. Making early assumptions about synergy between market segments can handicap the build-up of customer understanding. These kind of assumptions tend to pollute the model and inhibits clear and sharp reasoning.

many stakeholders are involved for any given customer. Multiple stakeholders are involved even when the customer is a consumer, such as parents, other family, and friends. Figure 4 shows an example of the people involved in a small company. Note that most of these people have different interests with respect to the system.

Figure 4: Which person is the customer?

Market segments are also still tremendous abstractions. Architect have to stay aware all the time of the distance between the abstract models they are using and the reality, with all unique infinitely complex individuals.
4 Life Cycle view

The basic CAFCR model relates the customer needs to design decisions. However, in practice we have one more major input for the system requirements: the life cycle needs. Figure 5 shows the CAFCR+ model that extends the basic CAFCR model with a Life Cycle view.

![Life Cycle View](image)

The system life cycle starts with the conception of the system that trigger the development. When the system has been developed then it can be reproduced by manufacturing, ordered by logistics, installed by service engineers, sold by sales representatives, and supported throughout its life time. Once delivered every produced system goes through a life cycle of its own with scheduled maintenance, unscheduled repairs, upgrades, extensions, and operational support. Many stakeholders are involved in the entire life cycle: sales, service, logistics, production, R&D. Note that all these stakeholders can be part of the same company or that these functions can be distributed over several companies.

References


History
Version: 0.4, date: June 30, 2010 changed by: Gerrit Muller