Apollo

Overview specification

Confidential

Revision number: 1

Date:

Name: P. van der Velde

# Disclaimer

This specification is not, by any stretch of the imagination, complete. It will need to be revised several times before it is complete. Currently several major parts are either missing or incomplete. This disclaimer will be updated to reflect any change in these sections. Finally a specification is supposed to be a ‘living’ document and therefore never complete. What were you thinking; better learn to live with this fact.

# Introduction

The design goal for Apollo is to provide the user, in general designers and engineers, with the capability to evaluate and change the response of geometry based designs on different physical situations. An example of this is an engineer who uses Apollo to determine the reactions of a design subjected to a set of external forces.

In order to achieve this goal the following targets for the design of Apollo have been set:

1. Provide an application that assists the user in any possible way with the task at hand without getting in the way.
2. Provide an application that can grow with the users demands for more complex designs.

All these targets are focussed on the user and the abilities the Apollo should provide for the user.

# Global overview

The Apollo application will consist of three different parts, the core, the user interface and one or more plug-ins. By separating out these three parts it is possible for the Apollo application to grow with the user by providing the desired functionality through one or more plug-ins. This allows additional functionality to be added (or removed) without directly affecting all of the application.

## Core

The main goal for the core is to provide the Apollo application with the basic utilities necessary to function. The core holds the kernel, responsible for the low level application functions, the project system and the user interface connections. The main responsibilities of the core are:

* Loading the different parts in the correct order upon application start-up and unloading them upon application shutdown.
* Provide a communication pathway between different parts of the application.
* Provide serialization capabilities for every component that needs to write information to a persistent store.
* Provide discovery and loading of plug-ins.
* Provide extension points for the user interface and the plug-ins.

Besides providing the infrastructure to achieve the before mentioned responsibilities the core also provides the connection points for the UI and the plug-ins. Separate API’s are available for both parts thus providing them with the ability to connect to the Apollo application.

## User interface

The main goal for the user interface is to provide the user of the Apollo application with a set of elements which allow control over the actions of the application. The user interface(s) connect to the core of Apollo via the User Interface Layer which resides in the core of Apollo.

The main goal of the user interface is to assist the user with the definition of the model and to translate the users requests in to concrete actions for the application. For instance when the user indicates that a certain region has a fluid flowing through it the user interface should translate this into the required set of physical models (fluid flow), boundary conditions (walls, inlet, outlet etc.), materials (e.g. air, water etc.) and components necessary to handle the users actions.

At the same time the user interface should be able to present the data that is stored in the datasets and compare different datasets in a comprehensive manner.

## Plug-ins

The plug-ins provide Apollo with all the data processing capabilities that it needs, both for the actual data generation and for the visualisation.