# PROPOSED SOLUTION

This dissertation proposes a new method for object-oriented framework design and instantiation.

In [19] Roberts and Johnson state that “Developing reusable frameworks cannot occur by simply sitting down and thinking about the problem domain. No one has the insight to come up with the proper abstractions.” They propose the development of concrete examples in order to understand the domain. The design strategy presented here is quite similar, analyzing concrete applications as viewpoints [8, 6] of a domain and deriving the final framework from the analysis of these viewpoints.

The approach to framework design is based on the idea that any design can be divided into two parts: the kernel sub-system design and the hot-spot sub-system design. The kernel sub-system design is common to all the applications that the framework may generate, and the hot-spot sub-system design describes the different characteristics of each application that can be supported by the framework. The hot-spot sub-system uses the method and the information provided by the kernel sub-system and may extend it.

The kernel structure is defined by analyzing the viewpoints design representations to produce a resulting design representation that reflects a structure that is common to all chosen viewpoints. This part of the design approach is based on a domain-dependent semantic analysis of the design diagrams to elicit the common features of the framework design structure, and is formally described in this dissertation.

The elements that are not in the kernel are the ones that vary for each application, and depend on the use of the framework. These elements define the framework hot-spots [17, 20] that must be adapted to each related application. We defined new relationship in object-oriented design, called the hot-spot relationship, to specify all the hot-spots in the system.

The semantics of this new relationship is given by the design patterns essentials [18]. This implies that the hotspot relationship is in fact a meta-relationship that is implemented through a design pattern that is generated taking into account the hot-spot flexibility requirements. The hot-spot cards guides this generation process, providing a systematic way for generating design patterns based on flexibility properties.

The most common way to instantiate a framework is to inherit from some abstract classes defined in the framework hierarchy and write the code that is called by the framework itself. However, it is not always easy to identify which code and where it should be written since frameworks class hierarchies can be very complex, especially for non-expert users.

Therefore, the “common” instantiation process is not always the ideal way to describe a particular application. This happens because of several facts:

* the language used to build and use the framework is a wide spectrum language, where it is not always easy to express the user intentions;
* the complexity of framework class hierarchies and the difficulty of finding the points where the code should be written, that are the framework hot-spots or flexible points.

The instantiation method proposes a different process, where a particular application is described by domain specific languages (DSLs) [12] that are designed precisely for the framework domain. In this way, the technique basic idea is to capture domain concepts in a DSL, which will help the framework user to build code in an easier way, without worrying about implementation decisions and remaining focused on the problem domain. The specification written in the DSL is translated to the framework instantiation code through the use of transformational systems [16, 4].