PROPOSED SOLUTION

In this dissertation, a fresh approach to designing and instantiating object-oriented frameworks is suggested.

"Developing reusable frameworks cannot occur by merely sitting down and thinking about the issue domain," Roberts and Johnson write in [19]. Nobody possesses the insight to create the right abstractions. They suggest creating real-world examples to help people comprehend the subject. A real application is analyzed as a viewpoint [8, 6] of a domain in the design process that is described here, and the final framework is then derived from the analysis of these viewpoints.

The kernel sub-system design and the hot-spot sub-system design are the two subsystem designs that make up the framework design process. The hot-spot sub-system design explains the various properties of each application that can be supported by the framework, whereas the kernel sub-system design is common to all the apps that the framework may generate. The kernel sub-approach systems and information are used by the hot-spot sub-system, which has the option of extending them.

The analysis of the perspective design representations results in a design representation that represents a structure that is shared by all the selected viewpoints, and this design representation is used to construct the kernel structure. This dissertation explicitly describes this component of the design strategy, which is based on a domain-dependent semantic analysis of the design diagrams to elicit the common elements of the framework design structure.

The components that depend on the usage of the framework and are specific to each application are those that are not included in the kernel. These components outline the hotspots in the framework [17, 20] that need to be

modified for each associated application. To specify every hotspot in the system, we created a new connection in object-oriented design called the hot-spot relationship.

The design patterns fundamentals [18] provide the semantics of this new relationship. This suggests that the hotspot relationship is a meta-relationship that is implemented using a design pattern that is created while considering the needs for hotspot flexibility. This generation process is guided by the hot-spot cards, which offer a methodical technique to create design patterns based on flexibility qualities.

Inheriting from some abstract classes defined in the framework hierarchy and writing the code that is called by the framework itself is the most typical technique to instantiate a framework. However, as class hierarchies in frameworks can be highly complex, especially for non-expert users, it is not always clear which code should be written.

Therefore, it's not necessarily best to describe a particular application using the "common" instantiation procedure. This occurs because of numerous facts:

- The complexity of the framework class hierarchies and the challenge of locating the places where the code should be written, which are the framework hot-spots or flexible points. The language used to build and use the framework is a wide spectrum language, making it not always simple to express the user intentions.
- The domain-specific languages (DSLs) [12] that are specifically created for the framework domain are used to describe a specific application in the instantiation technique, which suggests a different procedure. In this approach, the technique's primary goal is to record domain concepts in a DSL, which will make it simpler for the framework user to write code,

worry less about implementation choices, and stay concentrated on the issue at hand.

Using transformational systems, the specification written in the DSL is converted to the framework instantiation code [16, 4].