Integrated Model Refactoring Example

A simple example to illustrate the detection of the duplication model smell and application of model refactoring is provided in this section. This example can only show certain aspects of the usability of model refactoring. Since there is no visual representation of the integrated model, the examples include separate class, sequence and use case diagrams for comprehension. To make the presentation more concrete, we demonstrate the proposed refactoring application over a case study of Net Banking System (NBS). The following description sets up the context of the running example:

> *NBS is designed for financial institutions such as banks to provide their basic banking operations over the internet. The system allows customers to open accounts, perform online transactions like transferring money, paying bills and viewing account summaries. The system also allows bank operators and administrators to perform updates to the system online and handling other online operations.*

Figure 1 shows a subset of the model views from the NBS system that depicts the duplication model smell. On examination of the use case diagram, two paths associated with the actor *Operator* were identified. To ensure the existence of the model smell, the behavior of the middle use cases involved *Add Application Form* and *Edit Application Form* were observed. The sequence of message occurrence between the two interactions was found to be structurally similar. Hence, the existence of the duplication model smell was confirmed.

Initially, the *CreateUseCase (Manage Application Form)* refactoring is applied to create an empty isolated use case. Then the *ExtractFragment (Add Application Form, Manage Application Form)* refactoring is performed to copy the complete interaction fragment from one of the similar use cases (either can be used) into the new use case. To identify lexically different message interaction between the use cases, each message in the interaction of *Add Application Form* and *Edit Application Form* is compared. A message with a different name is replaced in the interaction of the new use case *Manage Application Form* with a new message. The following refactoring operation is hence applied *ReplaceMessage (createForm, manageForm(type))*. If both the messages are not used in any other interactions, they are replaced in the class diagram. The *MergeOperation (createForm, EditForm, manageForm)* refactoring is applied to the structural view to apply the change. Since the use of super-sub class relationship was not utilized (as lifelines in both the use cases were same), the *AddActorReference (Operator, Manage Application Form)* is applied. The duplicate use cases are initially isolated by applying the *IsolateUseCase (Add Application Form)* and *IsolateUseCase (Edit Application Form)* and finally deleted by applying the *DeleteUseCase (Add Application Form)* and *DeleteUseCase (Add Application Form).*
The refactored model views are shown in Figure 2.

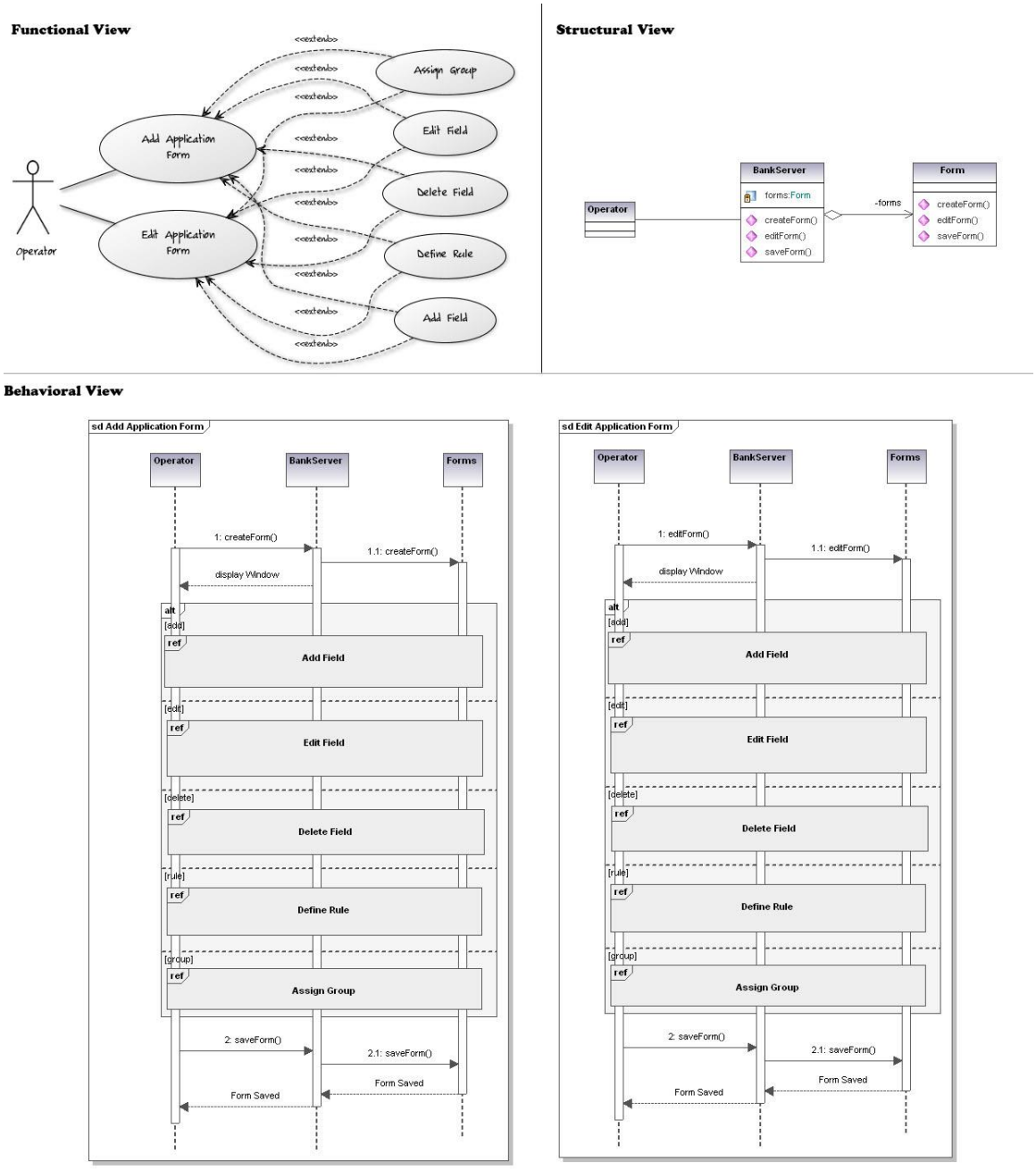

Figure 1 Excerpt of the NBS model views depicting Duplication Smell

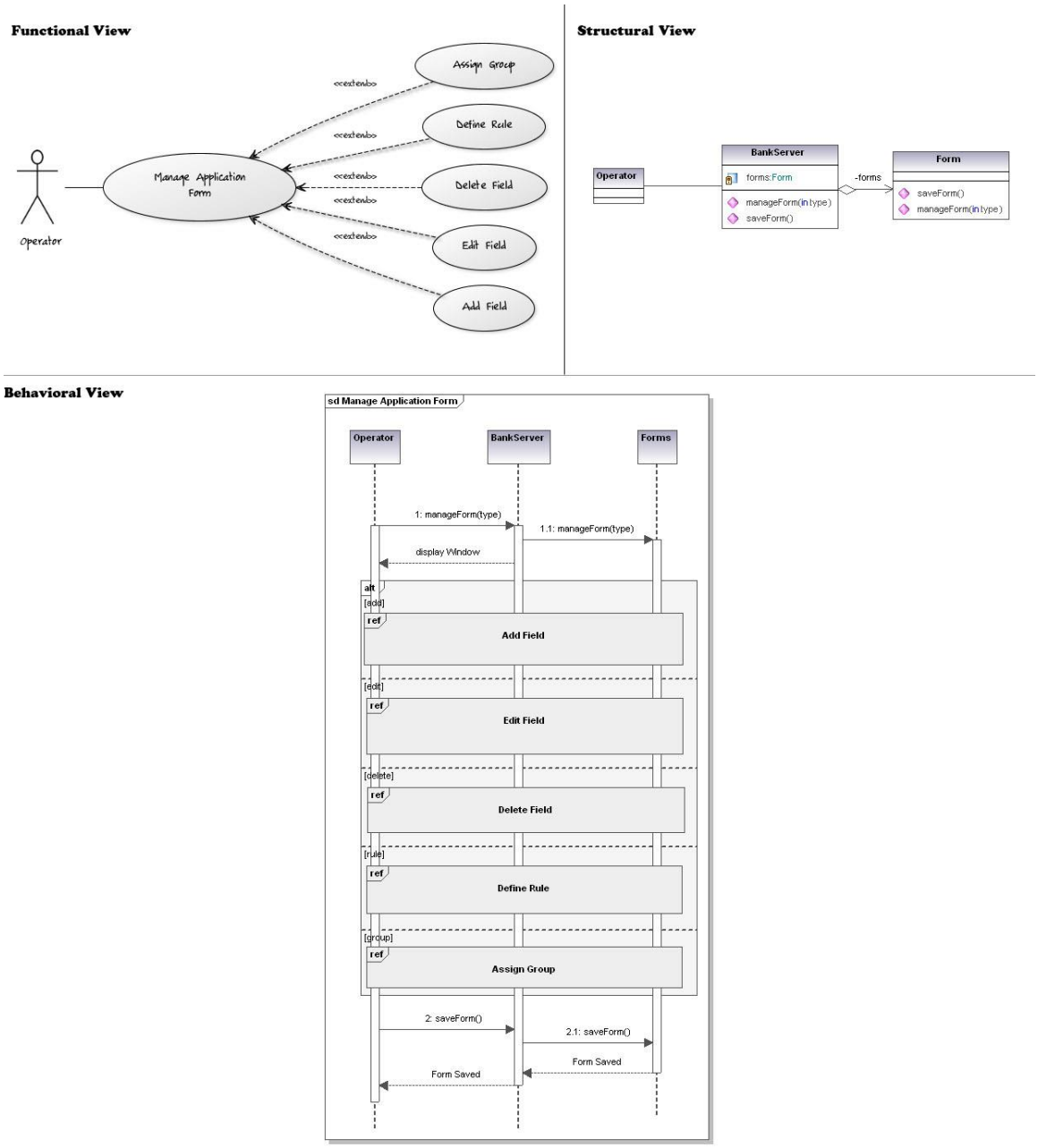

Figure 2 Excerpt of the NBS model views after refactoring