!! Sab Ka Malik Atma !!
Vishwatmak Jangali Maharaj Ashram Trust's
Atma Malik Institute Of Technology And Research

Client Management System

Submitted in Partial fulfilment of the requirement of the Degree of Bachelor of Engineering in Computer Engineering

By

Bhor Ajay

Kantoe Pratik

Mhaske Priyanka

Jadhav Devyani

!! Sab Ka Malik Atma !!
Vishwatmak Jangali Maharaj Ashram Trust's
Atma Malik Institute Of Technology And Research

DEPARTMENT OF COMPUTER ENGINEERING

Atma Malik Institute of Technology and Research

UNIVERSITY OF MUMBAI

2022-2023

!! Sab Ka Malik Atma !!
Vishwatmak Jangali Maharaj Ashram Trust's
Atma Malik Institute Of Technology And Research

CERTIFICATE

THIS IS TO CERTIFY THAT THE FOLLOWING STUDENT HAVE SATISFACTORILY CARRIED OUT PROJECT WORK ENTITLED **"Client Management System"** IS A BONAFIDE WORK OF SUBMITTED TO UNIVERSITY OF MUMBAI IN PARTIAL FULFILMENT OF THE REQUIREMENT.

Bhor Ajay

Kantoe Pratik

Mhaske Priyanka

Jadhav Devyani

Project Guide
Prof.Yogeshwari Hardas

head of department
Prof.Harshada Sonkamble

INDEX

1.Introduction

Customer data are one of the most important assets of every business. Effective storage and management of customer data is cornerstone of everyIT solution supporting company's business. It goes without saying that customer data are especially high-valued possessions for financial institutions and banking sector. In financial enterprise, it is equally important to collectinformation on customers as maintaining it and keeping it up-to-date. Furthermore, customer data have to be available for other parts of the complexIT solution where they need to be accessed or even modified. This bringsup several issues, such as data synchronization, that need to be addressed. This thesis will focus on customer data management of loan providing companies, where customer data is used in almost all processes from con- tract origination, client support to debt collection. A system of particular company was chosen for detailed analysis. The system also serves as reference model upon which all the improvements are designed.

1.1 Goal of thesis

Goal of the thesis is to describe processes related to customer data management in chosen financial institution that focuses on loan providing. Analysis of the current system used by the company, its functionality, interface and domain model.
Main goal is to provide improvements based on business needs as well as improvements of design shortcomings. Documentation of important use cases, designing of improved system with use of analysis patterns. Another goal is to describe system in context of other systems used in the company, how it fits in and the way it interacts with them.
Finally, create project documentation for implementing such changes. Documentation will contain project schedule, work effort estimation and risk analysis.

2.Customer management in financial company

This thesis is based on particular customer management system of Home Credit, originally Czech company founded in 1997, that started as a small loan provider for domestic appliances. In 2013 Home Credit is one of theleading providers of consumer loans in Europe and Asia. Home Credit Group (hereinafter referred to as the Group) is owned by PPF Group, one of the largest investing and financial enterprises in central and eastern Europe.Most of the software solutions for every Home Credit branch is pro- vided by Home Credit International (hereinafter referred to as HCI), a company that is part of Home Credit Group as well. Author of this thesis started working in the company on a position of system analyst in march 2012.

Customer information file:

Stand-alone system for customer management was first introduced in Home Credit branch in Russia. The system was named CIF which stands for Customer (or Client) Information File. Idea of the system is simple. To provide autonomous system that serves as master storage for all client related in- formation collected in contract origination phase and provide this information to other systems through defined interface. The interface allows for customer data to flow both ways so that outer systems are able to update customer data in CIF. More on CIF interface in separate chapter.

At the time of writing of this thesis CIF was being formed as a product. That means unifying object model and interface so that it can be easily deployed and fulfill business requirements of new founded branches. There will always be necessary customizations that need to be taken into consideration.

CIF in production environment:

It goes without saying that CIF as a stand-alone system does not bring any benefit to clients. Only with colaboration with other systems and applications does it bring benefits. It is part of software suite which is also being developed in-house by HCI. To describe the role of CIF in production environment we shall describe aforementioned suite, its purpose, features, architecture and components.

The Suite:

The Suite is a complex system for supporting business needs of Home Credit Group branches in the entire area of its operations. The word suite already hints that it comprises several blocks that together form a functioning system. Modular architecture of the Suite is the key feature that ensures high flexibility. It allows the Suite to be deployed in a new area of operation as well as partially or fully replace existing older systems. It gives us the possibility to answer actual needs of business by deploying only those modules that are truly needed to support those needs. For example, when business is being established in a new country, needs and product portfolio are not very complex as in country with successfully established business that is present for several years. But as the business grows, so do business needs and requirements change. Then it is time to deploy other modules of the Suite.

Vision of unified business processes is supported by unified architecture of the Suite. It is goal of HCI as a company to enforce members of Home Credit Group to share the vision of unified processes. It brings many ad- vantages. From the business point of view, major advantage is incorporat- ing experience from each area of operation to the whole Suite. Best practices from one area of operation can be transferred via the Suite to all the other areas. All members of the Group can contribute and at the same time benefit from improving the Suite for everybody.

Core of the Suite is based on stand-alone system for basic loan providing support. Connecting other components to this system led to establishing the Suite. The system had only few basic functionality in comparison to abilities of current Suite:

- Contract origination (loan application process)
- Contract life cycle management
- Loan approval
- Partner & sales network management
- Product management

Loan origination stands for creating an application for a loan by Home Credit employee or employee of the point of sale providing loans on behalf of Home Credit (e.g. electronic store). After approval of the

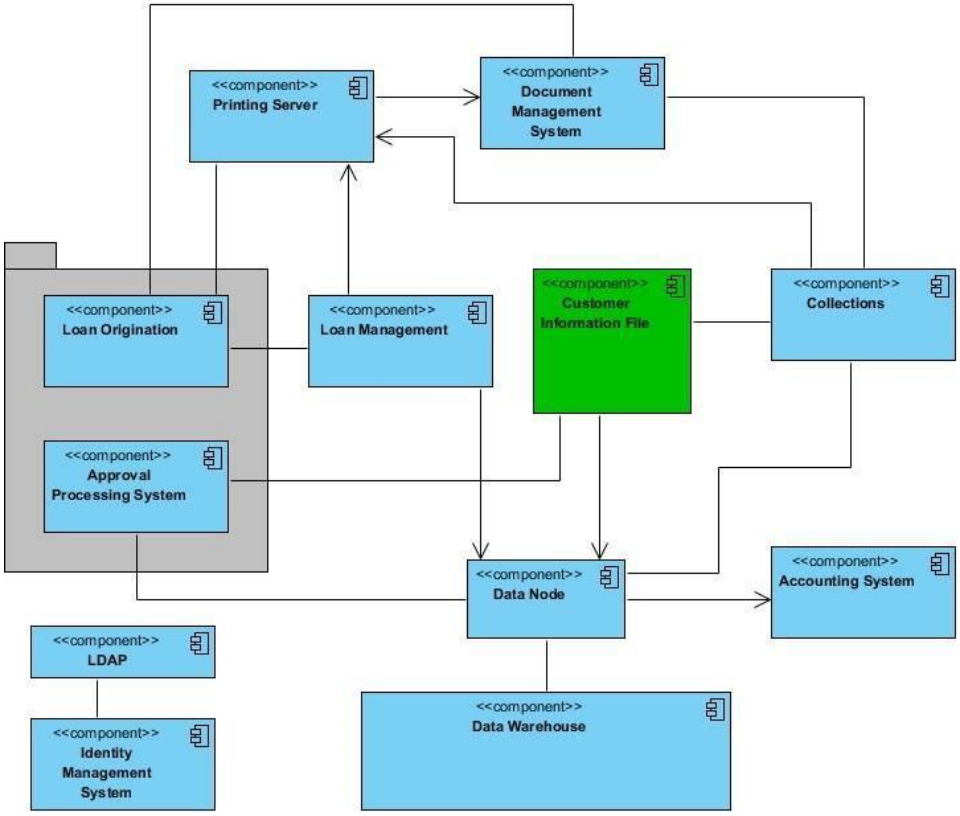


Figure 2.1: Suite architecture model

Besides contract management the system supports creating and managing of Home Credit partners and their network of salesrooms and points of sales (POS).

Loan application approval is ensured by separate module that works allows working with black lists, white lists and supports automatic and manual application processing activities.

The whole Suite comprise many other modules besides aforementioned core that supports business needs of group members. There is a module for card management used by group members offering credit card loans, server for printing documents and reports, document management system for storing documents and other file types. Module for campaign management and debt collection is also available and vital part of the Suite. There is also a data warehouse where crucial data is being replicated for

for accounting is also part of the Suite and used by some members of the Group.
Last but not least, modules for user management, authentication, authorization and overall security are part of the system. Some but not all applications that are part of the Suite support single sign-on feature of the authentication module. Single sign-on feature ensures that users working with more than one system does not have to sing in for each system separately but a common session is kept instead allowing him to access all systems with single login.

Role of CIF in the Suite:

Having briefly described the Suite itself and its components we can move on to Customer Information File. Purpose of the system is to store information about clients and provide them to other systems that work with them. Customer data enter the system during the phase of application origination. In the application customer's personal information is filled, some required and some optional. In the application general information like full name, date of birth, number of one or more identification documents (depends on what type of loan customer is applying for) can be filled. Additionally, contact information such as address, phone number, fax or e-mail addressis collected as well as more detailed information about his employment, financial status and even information about his relatives, colleagues or other persons related to the customer.
Data gathered from application are stored in queue processed by CIF. CIF performs identification algorithm with each customer data set that comes from the application. Client is then either successfully identified or new one is created in the system. Identification algorithm can be customized for needs of each Group member as the factors incorporated in the algorithm are based on culture and legal boundaries of the country where member of the Group operates. Client information input in the contract is also stored in order to keep data that were valid in the time of application origination. Client's data that are tied to given contract do not change. It is a job for CIF to maintain client data in and keep them up

Domain model:

Customer

Core entity of entire CIF system is Customer. Most of the attributes of Customer object are complex type attributes. Complex types are non-primitive objects with their own data structure. One of few simple attributes of Customer is CUID - Customer Unique Identification (ID). CUID is used as primary ID that unambiguously identifies an instance of Customer object and is also used by other systems or modules of the Suite for referencing to particular customer. There are several complex type attributes used in Customer such as Person Name, Country, Customer Birth, Customer nonymization ,Customer Personal Data, Customer Financial Data or Customer Un- desirability.

Person Name object contains several parts of name, salutation and honors before or after name. In many countries in which Home Credit Group members operate it is customary for citizens to have and use their middle name in official communications. Often it is the middle name that can differentiate two clients. In some countries father's full name is used too in official communications. That is the reason there are total of six attributesin Person Name for storing client's name.

Customer Birth object holds information on client's birth date and birth place, information about client's citizenship is stored in Country object.

Customer Personal Data contains more personal information about clients such as their education, marital status,information about family or residence.

Financial background of clients is saved in Customer Financial Data. It contains client's incomes, expenses and information about other loan payments.

Related persons:

One of extended information about customer is information about his relatives, sometimes referred to as contact persons. This kind of information has business importance, especially when it comes to *collection* phase. When client is late with his payments and direct communication channels are ofno use, information about contact persons - be it relatives, friends or col- leagues - is other way of contacting the client. Related Person uses some of the common complex types such as Person Name, Contact or Address. Relation to Contact and Address objects is bound via object Related Person Contact and Related Person Address respectively. Both mentioned objects have attribute role that states what kind of relationship does the related person have with client (parent, sibling, friend, colleague etc.)

Customer employment

Object Customer Employment contains information related to employment of clients. It includes information about client's salary, work experience, type of employment, client's profession and field he works in. Additionally, it contains another complex type attribute that is Employer. Employer consists of Employer Address and Employer Contact and is used in the same fashion as in Related Person. We can see that Contact and Address objects are generally used in relationship with many objects, keeping overall consistency of the model.

Enumerations:

Lot of information that is collected on clients has characteristics of sets that change very rarely or not at all. Information such as marital status, gender, education. Other information can be predefined for business needs such as type of address, type of contact on client, residence type and so on. All possible values can and in fact should have finite number of possible values. There can be many reasons for this. For example reporting, statistics, automated evaluation based on predefined criteria and so on. CIF uses collection of enumerations, so called "code lists" that are used for such attributes described earlier in this chapter.

Each code list has unique ID represented by integer (not a code as name would suggest). Identifying code list by integer has its roots in older system that is being systematically replaced in some countries. Purpose of code list can be known after reading its description. Each code list has several items that represent values used for certain attributes. Simplest example would certainly be code list 'Gender' that has of course two items: M (for male) and F (for female). Code list items have unique IDs, which are closer to codes than integer IDs of whole code lists. Item code is typically some string or single character such as gender items in previous example. The string is always UPPER_CASE with possible underscores, no white spaces are used. Code of this format has the advantage of carrying at least some in- formation value so when developer, analyst, or even customer (in this case, by customer we mean customer of HCI, not the customer of Home Credit) looks at it, for example in functional specification document, he is able in most cases determine what code list item that code represents. Values such as M, F or even better MALE and FEMALE have certainly greater meaning than values 0 and 1. However, integers were also used for code list items identifiers in older systems.

Code list items have more attributes than just code and name (or de scription), as can be seen in figure 2.5. Attribute order is for displaying items of code list on GUI in desired order, attribute visible can be used to hide given choice for user on GUI and attribute status states if the item is active or should be no longer used. Attributes min Elements and max-Elements represent cardinality of certain item. For example, there are allowed maximum of 3 and required minimum of 1 instances of contact

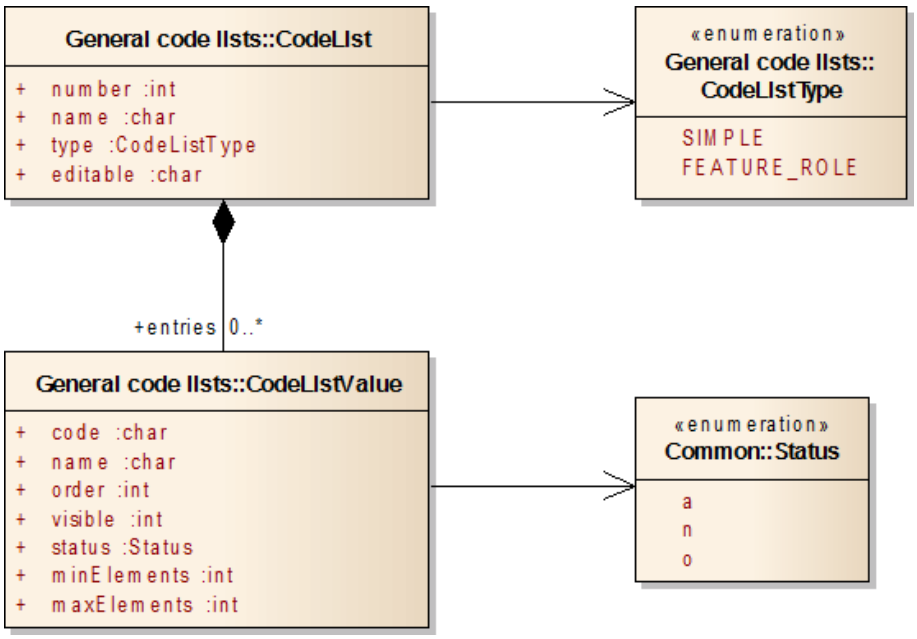


Figure 2.5: CIF general code list model

we need to filter subset of code list items based on other factor, for example by items from another code list? There are several cases when we need such functionality. Let us have two code lists - Industry and Profession. We need to "say" that certain subset of code list Profession is related to certain item in Industry. For example, profession items DEVELOPER, ANALYST and DB_ADMIN belong to industry item INFORMATION_TECHNOLOGY. There can also be need for deeper hierarchy, suchas document groups > document types > document attributes. There is no doubt that there may always be need to go deeper than that. Figure 2.6 shows model for so called document code lists that are used in CIF for purposes of storing code lists for document groups, types and attributes.
Unlike general code list model, we are able to capture relations and dependencies between more code lists. On the other hand the document code list model has strictly given purpose - it is used for given hierarchy of document groups, types and attributes with strictly specified attributes unqiue for those relations. In case of new business needs, e.g.: document attribute

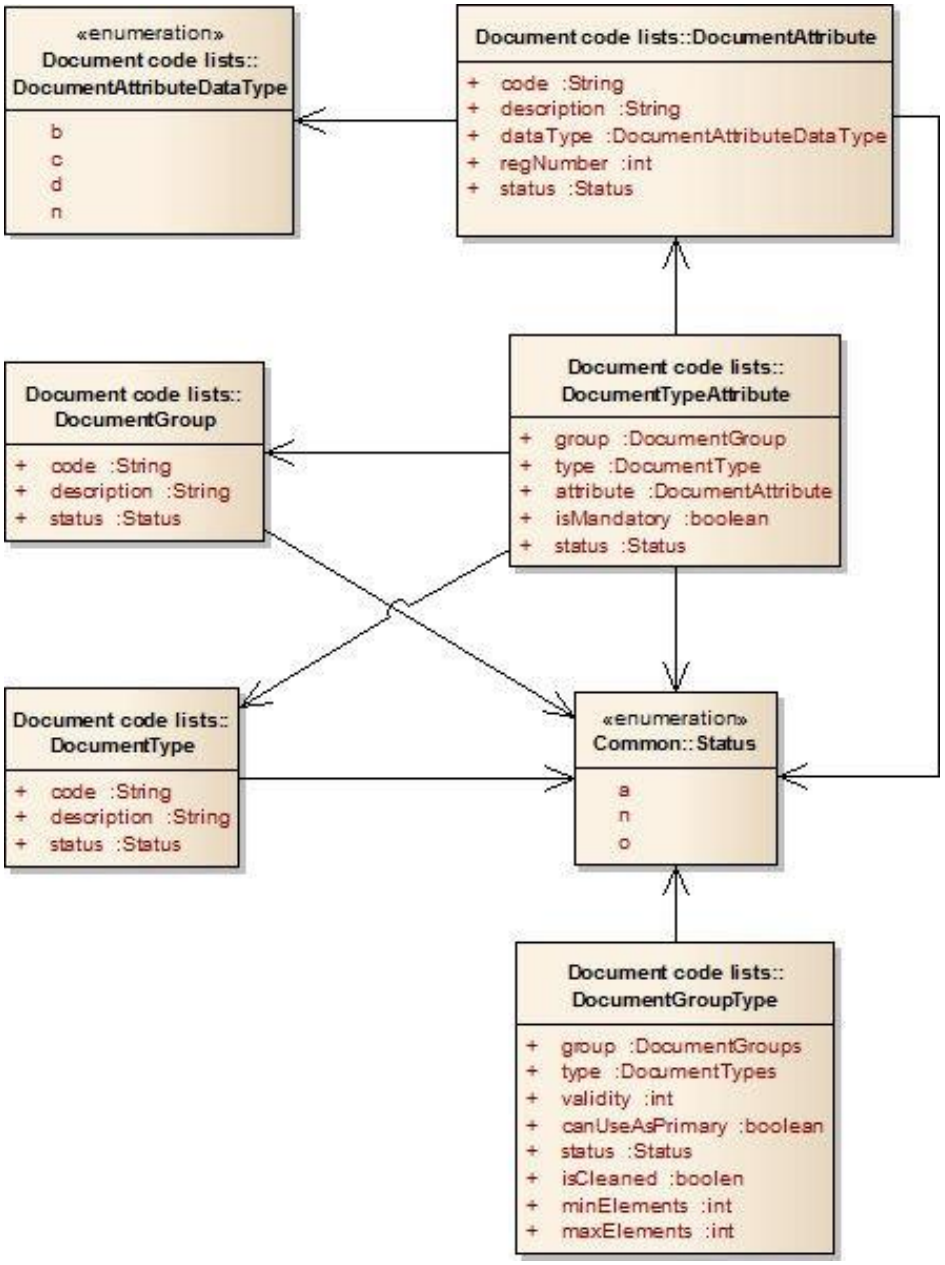


Figure 2.6: CIF document code list model

again, model needs to be changed. The same argument applies to

not so much flexible.
General structure needs to be introduced. Structure that will be able to support primitive code lists as well as more structured code lists with hierarchical relations and variable attributes.

Use cases:

Let us draw a simple use case diagram depicting basic use cases of the system. There are two actors on diagram: *Operator* and *System*. First mentioned actor represents user of system that does not necessarily have to be CIF. It can also be other system for implementing functionality for creating con-tracts. Such system can afterward send customer data via web service op-eration call to CIF. Each use case from use case diagram in figure 2.7 will be described in following subsections.

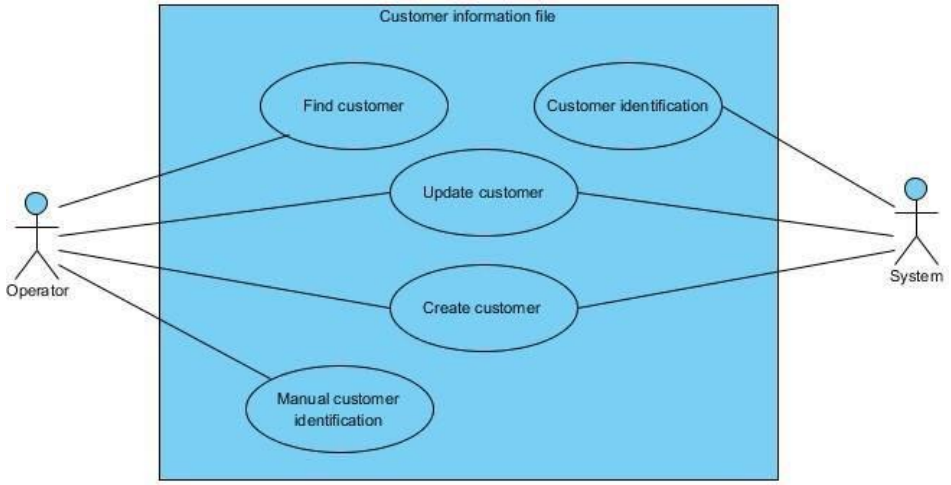


Figure 2.7: CIF - Use cases

Find customer:

Operator can find particular customer by providing criteria such as CUIDwhich is unique identifier of customer.

Update customer:

all information. This use case can be invoked for various reasons:

- Operator that created contract with client finds out he made a mis-take and wants to fix it.
- Client calls and informs about mistake in contract. Operator wantsto fix the mistake.
- Client himself informs about change in his personal information (e.g.his contact address or phone number changed).
- Operator in collections department tries to reach client and finds outthat he is not reachable on one of his addresses or phone numbers.
  Such obsolete data can then be deactivated by operator.

Create customer:

Use case can be executed by operator as well as system itself:

- When sending customer data to system, customer identification is executed. One of the outcomes is creating new customer record in the system.
- When operator performs manual customer identification, he can de-cide to create new customer record in system.

Customer identification:

This use case is executed by system when it processes customer record sent from outside (e.g. after creating contract). System launches algorithm that basically tries to find customer in its own database based on defined crite- ria. If the system finds a match it goes deeper and tries to determine if found customer record is really the one it is now processing. If a match is even- tually found system replaces data of existing customer with data of pro- cessed customer. If customers did not fully match or if even more than one match was found in the first place, all these are potential duplicate records. Processed customer record is put into queue for manual identification and operator must manually process this customer record.

Manual customer identification

When record of customer is forwarded to queue for manual identification

face is opened for the operator where he can see vital customer data includ- ing customer's photo on one part and on the other portion of the screenhe can see information about one of the potential customer duplicates. Op- erator can go through the information and compare the customer records. Finally he makes a decision:

1. Customer does not match any of the potential records that exist in the system and new record is created.
2. Customer record is actually one of the existing ones and his informa- tion is updated.
3. He rejects customer. There can be several reasons, for example customers differ in names but number of their identification documentis the same based on the photos, they are clearly different persons. In that case operator can reject the new customer under suspicion of fraud.
4. Operator is not able to make a decision at the moment so he returns customer to the queue. Returned customer recored can be picked up later by another random operator.

Interface

In chapter 2.2.2 we have described role of CIF as master system for clientdata. On diagram depicting architecture of systems including CIF (figure 2.1) we are able to see that CIF is definitely not an isolated system. On thecontrary, its purpose is to communicate and exchange client information with different systems.
Technology used for achieving system interoperability is Web Services Architecture[1], a standard that is platform and framework independent. Because Java is primary technology used for development of most of the systems in the Suite, CIF included, Java Web Services implementation is used.It relies on XML standards and transport protocols to carry the data from system A to system B.
CIF system has several Web Services (WS) deployed for particular area of the system such as customers or code lists. Each WS implements several operations that can be called remotely by accessing known WS endpoint. For obvious reasons thesis does not provide exact description of WS opera- tions or WSDL[2]. Instead we will outline operations in a nutshell.

3.Analysis

In this chapter we shall analyze CIF system as it is and create its analyticalmodel that will correspond with current abilities of our system. We will then continue to make improvements of the model in order to make it more versatile and ready for possible need of functionality extension that can always arise from business requirements.

1.2 Introduction to methodology

For the purpose of analysis we will use Unified Modeling Language (UML), a well established, standardized language for multipurpose modeling in the field of software engineering. We will use mainly class diagrams for modeling analytical model of our system. UML is often mentioned in con-text of Unified Process (UP). Unified process, which is commonly used term for methodology called Unified Software Development Process, is an indus- try standard of Software Engineering Process created by authors of UML.Unlike UML, UP is not part of OMG[1] standards. [ARLOW, OMG]

1.3 System requirements

We have introduced and described CIF system in chapter 2. Before we start analyzing and creating class and use case models of the system, let us formally introduce requirements for the system. Requirements that will be described later in this section will be in accordance with the current system functionality. Additional requirements will be set continuously later in this chapter and model improvements will be done so that those requirements are met.
Primary system requirement is for the system to be able to store client information:

Each client can have many related persons but only one employment and employer. Address structure must be able to be used for addresses in different countries where address formats vary. Contact types have to be differentiated, for example phone, e-mail, web site or fax.

Other requirements:

- User defined enumerations
  - System must support user defined enumerations that can beused as type for specific attributes, e.g. gender (male, female) or family status (married, single, etc.).
  - Each enumeration and enumeration item must have unique codesso we can use them as references but more importantly to be able to synchronize these enumerations between systems.
  - Each enumeration item needs to have status information: *ACTIVE*, *INACTIVE*, *OBSOLETE*. This helps us to switch status of individual item via configuration. To stop accepting certain enumeration value we can just set the proper status without deleting the value completely. Active and inactive statuses areself explanatory. Idea behind status value *OBSOLETE* is that the systems which use certain enumerations should stop offering items with status *OBSOLETE* as selectable options on user interface

Analysis classes:

Before we begin work on analysis model using UML class diagram, we shall introduce and describe briefly core analysis classes that will be used laterin modeling phase.

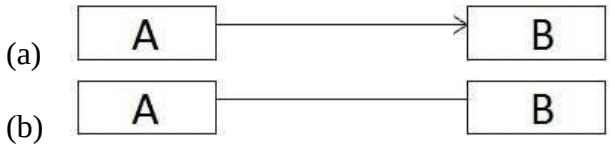
Analysis classes should be based on real life entities from the target domain. Analysis class ideally maps real life business terms such as "cus- tomer", "product" or "account". This statement applies under condition that business can be clearly defined and unambiguous which is rarely the case.In analysis phase only a subset of full UML syntax is used. More depth anddetail is added to classes during design phase that comes after analysis. Therefore, analysis class must contain at least name and several attributes. As for the attributes we can assume that analysis model will almost nevercontain full or final list of attributes, rather attribute candidates. Attributes will most likely be modified during design phase.

Problem of finding analysis classes can be approached from different sides. Noun verb analysis is well known approach in object oriented anal- ysis and design. This process allows us to identify key classes and their behavior from nouns and verbs in story about what the system is and what it does. CRC (class, responsibility & collaboration) cards are brainstorm- ing tools and another way to approach analysis classes identification. This methodology consists in writing physical paper cards that represent classes. Each card contains class name, responsibilities of the class and names of other classes that the class will collaborate with. CRC cards are popular be- cause people from business and people with no IT background in general can be involved in the process. According to Rational Unified Process (RUP) methodology analyst should think of three different types of analysis class, each marked with stereotype:

Now that we have described major analysis classes we can move on to anal- ysis model where we will use those classes and connect them with associ- ations, another UML concept. Associations are special type of relations be- tween classes. Similar to associations are links between objects which are concrete instances of classes. Semantic of association is fairly simple. As- sociation merely means that we are able to create links

between objects of associated classes. There are advanced relations that are improvements of associations: aggregation and composition. Relations aggregation and com- position are mostly used in design phase where associations are being re- placed by compositions and aggregations. [ARLOW]

Since we are in analysis phase following model will contain simple associations only. When we look at the model we can see with no surprise that *Customer* is associated with practically each class except *Employer*. *Employer* is rather associated with *Employment* and according to association multi- plicity it is optional. Same goes for *Employment* that is optionally associated with *Customer*. Classes *Address* and *Contact* are associated with *Customer*,



We will be using standard practice idiom. That implies all associations used in model in figure 3.11 are navigable in both ways. It is not significant to specify navigability in such early analysis phase.
Figure 3.12 shows model of code lists - general user defined enumerations we have mentioned earlier. Model contains two classes: *CodeList* and *CodeListItem* with one enumeration *Status*. Model is fairly simple and idea behind these classes have been explained in the respective subsections.

Now that we have basic analysis model, let us go deeper into certain classes and see if we can make parts of the model more specific. We shallexamine particularly *Customer* and *Document* class.

Splitting Customer class:

If we look at requirements for type of customer data that our system should collect we can see that they have been divided into specific categories: general data, personal data, financial data, client undesirability and client anonymization. In the basic model we defined attributes from all these areas. Although
the information is specific for one client, we can extract attributes specific for each area into new class. This decision has few advantages. In spite of creating more classes in diagram it is easier to find relevant information compared to going through one class encapsulating all the attributes. Furthermore, we may not need all the attributes every single time. In some

cases we will need only core client information without his financial or family background. If we compartmentalize client information into more classes we can always load only those parts that we need in specific situation.

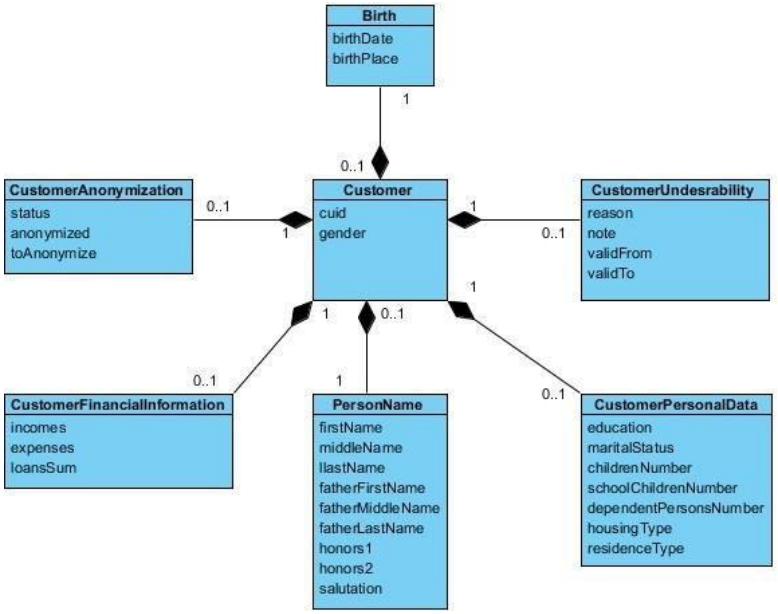


Figure 3.13: Split Customer class

After extracting financial and personal data from *Customer* class we get several new classes as can be seen in figure 3.13. *Birth* and *PersonName* were previously attributes in *Customer* class but now we have expanded them be- cause their type, although we do no model types in our models, is not sim-ple primitive. We do not want to store customer's name in a single string but instead save each part of name in one primitive type attribute. Also, in real world, parts of name are often person's titles and honors that can be written before or after person's name. That is why we have also attributes *honors1* and *honors2*. Finally, attribute *salutation* can be useful in official let- ters or in other forms of communication with client.

Classes are associated with *Customer* class with special type of associa- tion - composition. Composition is stronger type of aggregation which is as- sociation of type part - whole. It is transitive and asymmetric

Composition was chosen instead of aggregation from following reason: parts of *Customer* cannot exist alone. They are always created and destroyed with *Customer*.

3.4.1 Extending Document model

In our first analysis model (figure 3.11) of CIF system we have simple class *Document* that among other attributes has attribute *group* and *type*. In this section we will develop this concept in more detail. In figure 3.14 is class model depicting document group, type and attribute structures. Essentially document group has document types, document type can have attributes specific to document type.

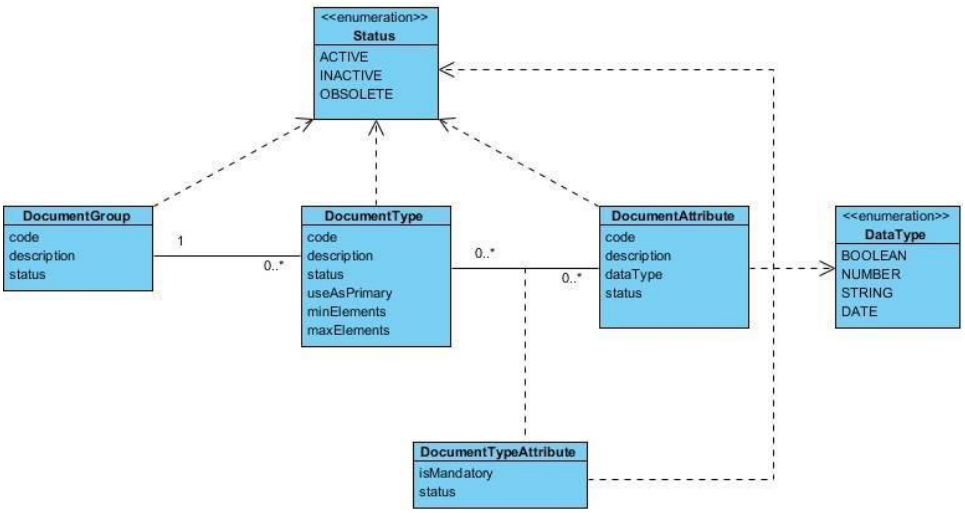


Figure 3.14: Extended Document model

Document Group class has few essential attributes: code, description and status. It has association with Document Type class with multiplicity 1:N. We assume that each document type belongs to precisely one document group. Document Type possesses the same core attributes as Document Group,but also other attributes. Document Attribute class represents attribute that can be defined for certain types of documents and of course document type can contain many such attributes. This is deducible from many-to-many relationship between Document Type and Document Attribute. Relationship between these two classescan have attributes unique to the relationship only. This is classic

tributes is Mandatory and status. It may seem that status attribute is redundant because same attribute is part of Document Type and also Document Attribute. Attribute status has its purpose in association class Document Type Attribute. By setting INACTIVE status for particular attribute we make use of the attribute invalid overall. But setting this value for status in association class we can invalidate usage of that attribute only for related document type.

Putting it together:

By putting some of the features discussed in previous sections together we get one more or less complete model (figure 3.15) of our system that should be able to fulfill defined requirements. System model allows us to capture general information about clients as well as information from different areas such as finance and family background. We are able to store information when client is labeled as undesirable and we can save anonymization status. Model enables us to save other vital parts related to client: his contact information, address, his consents, documents, employment and related persons.

The next chapter will be about modifying this model for better or more general use. We will also modify or define completely new requirements that the new model will have to meet. We will be drawing inspiration from analysis patterns that have already been created for recurring analaysis problems in various business areas.

4.Model improvements

Previous chapter was dealing with creation of analysis model for Customer Information File system. We started by defining requirements for the system, continued with finding core analysis classes and continued by modeling small portions of the domain which ultimately led us to completing the whole model. We are not implying the model has exactly the same capabilities to the last attribute of the actual system used by Home Credit branches. This was never the intention but rather take the key aspects of the actual system and build upon them an improved version and broaden its capabilities.

If we look closely at the system we can spot several shortcomings or restrictions. This is natural because as it was stated, model of the system is based on actual system of actual company and the development of that system was based on real business requirements of real clients. For instance, one of the more obvious restrictions is that the system is person centric and does not take legal entities as clients into consideration. The system could be misused or forced to work with such type of client but such an approach would not be very clean and suitable. This restriction that was just mentioned has also its roots in needs and nature of Home Credit business. Since the company focuses on personal cash and consumer loans and not on loans for firms and other legal bodies it is understandable and completely justified decision. *"You need the simplest model you can get away with. Don't add flexibility you are unlikely to use."* [FOWLER]

On the other hand CIF system is used not only by Home Credit branches but Air Bank as well since it is member of PPF Group. We already discussed core business of Home Credit. Bank typically offers standard products such as variety of bank accounts, mortgage and loans. Now banks typically provide loans not only to natural persons but also to legal entities - companies, organizations or governments. This chapter will be focused on modifying our current CIF model in order to support evidence of clients in general - be it natural persons or legal entities - and other modifications that will make the system more flexible.

Generalization

In this section we will try to find some common features of CIF entities and use generalization to capture them into superior entity. In our case there is more than one option how to apply generalization.

From the system analysis we know that besides customer and newly added legal entity also employers and related persons can have contacts and addresses. We could add *RelatedPerson* and *Employer* to "party" model in figure 4.4 as children of *Party* on the same level as *LegalEntity* and *Per- son*. Result can be seen in figure 4.5. But could we find any common features that are shared by the entities? Let us find out.

5.Project documentation

In this chapter we will provide basic project documentation consisting of risk analysis and project plan.

Risk analysis

When introducing any kind of innovation into business, the company must take into account possibility of various types of risks that may cause troubles like inefficiency of the new method, higher costs and lower profit than expected, etc. Thus risk analysis is frequently used when planning a project to help the company management decide whether it is a good idea to change the old routine and if arising profits are worth the possible risk. Results of risk analysis can be used to foresee the most possible risks coming with the newly planned project, may it be on internal or external site. Most frequently internal risks are technology failure, staff fluctuation - due to sickness or financial situation and lowering the budget for new project due to (change in) management decisions. External risks are for example instability of cash flows financing the new project due to unpredictable changes in stock and financial markets, decline in demand of the offered product in the public and even extreme events in form of natural disasters. The latter can also indirectly cause the previous.

Identified risks

We have identified following risks arising from implementation of changesto system analysis model proposed in previous chapter:

- Lack of resources We may encounter risk of lack of required re-sources in terms of manpower to complete or even start the project. Company may be overwhelmed by by projects with higher prioritythat need to come first. Project may be postponed indefinitely.
- Too great impact on surrounding systems. Assessment of impact of CIF redesign on surrounding systems may prove to be too great and the changes not worth the effort.
- Stakeholders not interested. Deadly risk. Project stakeholders needto be convinced that changes though not producing any benefits im-

ture implementations of system. If stakeholders are not on board the project is endangered.

- Lack of finances. Standard risk needs to be considered almost inany project. Directly influenced by previous risk (Stakeholders not interested). Without stakeholders approval they will not finance the project.

Project plan:

Project plan comprises list of tasks (figure 5.1), project Gantt chart (figure 5.2) and also list of resources (figure 5.3) that are part of task list. Plan starts with creating new analysis model of system (CIF) and continues with creating of design model that is based on new analysis model. It involves other tasks concerning database modification, data migration, unit testing and integration testing.

Plan has been devised for implementation of modifications of CIF system only. Changes include modification of XSD schema used in Web Ser- vices. This change brings necessary modifications on side of systems that communicate with CIF via WS interface. It would take detailed analysis ofeach system and what WS each system actively uses in order to determine impact on the system and estimate costs of required modifications.

	Name	Duration	Start	Finish	Predecessors	Resources
0	**⊟CIF modification**	**73d**	**09/02/2013**	**12/11/2013**		
1	Modification of current CIF analysis model	10d	09/02/2013	09/13/2013		Analyst 1,Analyst 2
2	Specification of new design model based on new analysis model	10d	09/16/2013	09/27/2013	1	Analyst 2,CIF domain leader
3	Source code rewriting according to design model	30d	09/30/2013	11/08/2013	2	CIF domain leader,CIF developer 1
4	Modification of XSD model for Web Services	12d	09/30/2013	10/15/2013	2	CIF developer 2
5	New data model design	5d	09/30/2013	10/04/2013	2	CIF database specialist
6	Creating database modification & migration scripts	8d	10/07/2013	10/16/2013	5	CIF database specialist
7	Deployment of new DB shcema & data migration	2d	10/17/2013	10/18/2013	6	CIF database specialist
8	Unit testing	3d	11/11/2013	11/13/2013	3	CIF developer 1,CIF developer 2
9	Deployment to testing environment & configuration	2d	11/14/2013	11/15/2013	7,8	Operations department
10	Integration testing	18d	11/18/2013	12/11/2013	9	Testing department

Figure 5.1: Project plan task list

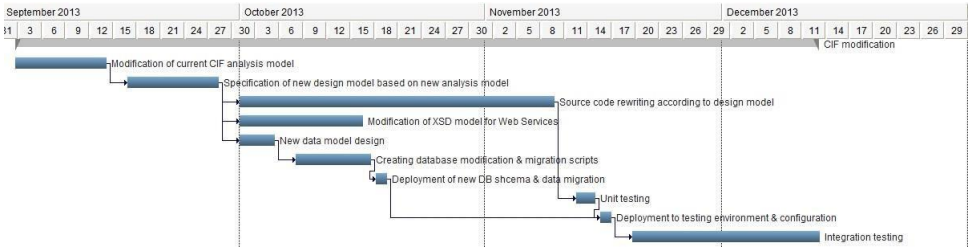


Figure 5.2: Gantt chart

	Resource Name	Type
1	Analyst 1	Work
2	Analyst 2	Work
3	CIF domain leader	Work
4	CIF developer 1	Work
5	CIF developer 2	Work
6	CIF database specialist	Work
7	Application architect	Work
8	Testing department	Work
9	Operations department	Work

Figure 5.3: Project resources

6.Conclusion

Main goals of the thesis were to describe significant processes related to loan providing in chosen financial institution, to analyze current system, describe important use cases, interface and model of the domain. Furthermore, to provide analysis model of the system, describe its limitations and come up with solutions to redesign the model so that it can be adapted by other companies from financial sector.

In the second chapter we have introduced loan providing company Home Credit and described the domain of Customer Information File system. Wehave introduced domain model with examples of XSD diagrams, use cases and briefly Web Service interface.

Third chapter provides analysis of the current system starting with defin- ing formal requirements on system's capabilities for storing client informa- tion. Followed by identifying analysis classes we finally created analysis model of the system part by part.

Fourth chapter is about improving the model from chapter three. It gives the reader an introduction to patterns followed by specific pattern descrip- tion that is used to redesign part of system model. This part of the model is further modified with by using generalization concept that is discussed in detail. Multiple inheritance phenomenon that can arise by using this concept is described as well as means to dealing with it in regard to programming languages that do not generally support the concept of multiple inheritance.

In fifth chapter we provide risk analysis consisting of identifying of key risks that could influence realization of the project which purpose would be to implement changes to the system model. Project plan is included as well comprising task list, Gantt chart and resources needed for realization.

Result of the project is an analysis model of system for storing client information in financial institution that is based on particular system developed by HCI company and used by Home Credit branches. We have provided some proposals to redesign the model based on potential needs of business in financial sector and to increase flexibility. Primary goal wasto provide complete redesign of the system with help of suitable analysis patterns that have been discovered and proven in actual projects and pro- vide completely redesigned model.