SYSTEM ANALYSIS AND DESIGN OF BRAC MIACRO CREDIT PROGRAM

A Thesis

Submitted to the Department of Computer Science and Engineering

Of

BRAC University

By

Md. Obaidul Kabir

Student ID: 01101018

RAZWANUL HUQ

Student ID: 01101001

In Partial Fulfillment of the

Requirements for the Degree

Of

Bachelor of Science in Computer Science

May 2005

DECLARATION

I hereby declare that this thesis is based on the results found by myself. Materials of work found by other researcher are mentioned by reference. This theis, neither in whole nor in part, has been previously submitted for any degree.

Signature of
Supervisor

Signature of
Author

Acknowledgement

After finishing our report, we acknowledge all the persons who have interacted with us. Without their help, this report writing would have been absolutely impossible for us. So our sincere thanks to them for their cooperation and support.

We would like to thank Mr. Manzur Ashraf, our honorable Supervisor for guiding and helping to make this thesis report. This thesis gives us the chance of working with real life data, which helped us to acquire clear insight about the real life systems. It was a thoroughly enjoyable experience. We also want to thank each and every person who helped us in different way to the research procedures and timely accomplishment of the report.

Our special gratitude and appreciation goes to Mr. Sabbir Ahmed Chowdhury, Program Head, Micro credit department, BARC. The cooperation and assistance of the managerial staff of these organization are greatly appreciated .we are really grateful to them for their assistance and guidance that they have given in collecting data. We are thankful to all of my classmates- for continuous support, assistance, and inspiration.

However, we are the only responsible for the errors and omissions in the report, if any.

ABSTRACT

The goal of the thesis is to develop a software tool, which is capable of storing all the information of loan holder and loan giving scheme, the criteria and constrains of giving loans. We plan to focus on analysis and design of the existing system. A user provides the data of the members along with specifications. The software is responsible for finding a solution. The difficulty of this research project is in designing an effective method to resolve the entire loan scheme and to come up with a properly compensated system

CHAPTER I

INTRODUCTION

COMPUTER IS A MACHINE THAT PERFORMS TASKS, SUCH AS CALCULATIONS OR ELECTRONIC COMMUNICATION, UNDER THE CONTROL OF A SET OF INSTRUCTIONS CALLED A PROGRAM. PROGRAMS USUALLY RESIDE WITHIN THE COMPUTER AND ARE RETRIEVED AND PROCESSED BY THE COMPUTER'S ELECTRONICS. THE PROGRAM RESULTS ARE STORED OR ROUTED TO OUTPUT DEVICES, SUCH AS VIDEO DISPLAY MONITORS OR PRINTERS. COMPUTERS PERFORM A WIDE VARIETY OF ACTIVITIES RELIABLY, ACCURATELY, AND QUICKLY.

There are not only a variety of uses of computers in systems, but there are also many different ways to build systems. Sometimes a system just evolves .It may start with a very simple program to which more and more functionality is gradually added. Sometimes systems are built experimentally, trying out new ideas as the system is built. On the other hand, the development of an information system for a large business application must be carefully planned and monitored. Many information systems now use computer systems for manipulating information and are sometimes called computer-based information systems.

1.1 System

SYSTEM IS ANY COLLECTION OF COMPONENT ELEMENTS THAT WORK TOGETHER TO PERFORM A TASK. IN COMPUTER SCIENCE, SYSTEM IS USED IN A VARIETY OF CONTEXTS. A COMPUTER IS A HARDWARE SYSTEM CONSISTING OF A MICROPROCESSOR AND ALLIED CHIPS AND CIRCUITRY, PLUS AN INPUT DEVICE (KEYBOARD, MOUSE, DISK DRIVE), AN OUTPUT DEVICE (MONITOR, DISK DRIVE), AND ANY PERIPHERAL DEVICES (PRINTER, MODEM). WITHIN THIS HARDWARE SYSTEM IS AN OPERATING SYSTEM, OFTEN

CALLED SYSTEM SOFTWARE, WHICH IS AN ESSENTIAL SET OF PROGRAMS THAT MANAGE HARDWARE AND DATA FILES AND WORK WITH APPLICATION PROGRAMS. EXTERNAL TO THE COMPUTER, SYSTEM ALSO REFERS TO ANY COLLECTION OR COMBINATION OF PROGRAMS, PROCEDURES, DATA, AND EQUIPMENT UTILIZED IN PROCESSING INFORMATION: AN ACCOUNTING SYSTEM, A BILLING SYSTEM, A DATABASE MANAGEMENT SYSTEM.

SYSTEM IS AN INTEGRATED AGGREGATION OF END ITEMS, INTERFACES, AND SUPPORT FUNCTIONS DESIGNED TO FULFILL A SPECIFIC MISSION REQUIREMENT. A SYSTEM MAY INCLUDE EQUIPMENT, TRAINED PERSONNEL, FACILITIES, DATA AND PROCEDURES, AND SOFTWARE. FOR PROJECT PURPOSES, A SYSTEM IS TYPICALLY DEFINED AS THE HIGHEST LEVEL OF HARDWARE ORGANIZATION COMPOSED OF MULTIPLE SUBSYSTEMS. THE TERM IS ALSO USED TO DESCRIBE A DISCIPLINED AND CONSISTENT APPROACH TO ACCOMPLISH A TASK.

1.1.1 System Analysis and Design

SYSTEM ANALYSIS AND DESIGN IS A COMPLEX, CHALLENGING AND SIMULATING ORGANIZATIONAL PROCESS THAT A TEAM OF BUSINESS AND SYSTEMS PROFESSIONALS USES TO DEVELOP AND MAINTAIN COMPUTER-BASED INFORMATION SYSTEM.

SYSTEM ANALYSIS IS "FINDING OUT WHAT A SYSTEM DOES AND WHAT ITS NEEDS ARE". SYSTEM ANALYSIS IS AN IMPORTANT ACTIVITY THAT TAKES PLACE WHEN NEW INFORMATION SYSTEMS ARE BEING BUILT OR EXISTING ONES ARE CHANGED. ITS MORE CRUCIAL ROLE IS IN DEFINING USER REQUIREMENTS. SYSTEM ANALYSIS, OFTEN NOW CALLED BUSINESS SYSTEMS ANALYSIS TO EMPHASIZE ITS BUSINESS EMPHASIS, IS NEEDED IN THE FIRST INSTANCE TO CLEARLY IDENTIFY WHAT IS POSSIBLE AND HOW A NEW SYSTEM WILL WORK. THIS INCLUDES GATHERING THE NECESSARY DATA AND DEVELOPING MODELS FOR NEW SYSTEMS.

INFORMATION SYSTEMS ARE DEVELOPED FOR DIFFERENT PURPOSES, DEPENDING ON THE NEEDS OF THE BUSINESS. TRANSACTION PROCESSING SYSTEM (TPS) FUNCTION AT THE OPERATIONAL LEVEL OF THE ORGANIZATION; OFFICE AUTOMATION SYSTEM (OAS) AND KNOWLEDGE WORK SYSTEM (KWS) SUPPORT WORK AT THE KNOWLEDGE LEVEL. HIGH-LEVEL SYSTEMS INCLUDE MANAGEMENT INFORMATION SYSTEM (MIS) AND DECISION SUPPORT SYSTEM (DSS). EXPERT SYSTEM APPLIES THE EXPERTISE OF DECISION MAKERS TO SOLVE SPECIFIC, STRUCTURED PROBLEMS. ON THE STRATEGIC LEVEL OF MANAGEMENT WE FIND EXECUTIVE SUPPORT SYSTEM (ESS). GROUP DECISION SUPPORT SYSTEM (GDSS) AND THE MORE GENERALLY COMPUTER SUPPORTED COLLABORATIVE WORK (CSCW) SYSTEM AID GROUP-LEVEL DECISION MAKING OF A SEMI STRUCTURED OF UNSTRUCTURED VARIETY.

1.2 Systems Development Life Cycle

Systems Development Life Cycle (SDLC) is the overall process of developing information systems through a multi step process from investigation of initial requirements through analysis, design, implementation and maintenance. There are many different models and methodologies, but each generally consists of a series of defined steps or stages. The Systems Development Life Cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project from an initial feasibility study through maintenance of the completed application. Various SDLC methodologies have been developed to guide the processes involved including the waterfall model (the original SDLC method), rapid application development (RAD), joint application development (JAD), the fountain model and the spiral model. Mostly, several models are combined into some sort of hybrid methodology.

In general, an SDLC methodology follows the following steps:

1. The existing system is evaluated. Deficiencies are identified. This can be done by interviewing users of the system and consulting with support personnel.

2. The new system requirements are defined. In particular, the deficiencies in the existing system must be addressed with specific proposals for improvement.

3. The proposed system is designed. Plans are laid out concerning the physical construction, hardware, operating systems, programming, communications, and security issues.

4. The new system is developed. The new components and programs must be obtained and installed. Users of the system must be trained in its use, and all aspects of performance must be tested. If necessary, adjustments must be made at this stage.

5. The system is put into use. This can be done in various ways. The new system can phased in, according to application or location, and the old system gradually replaced. In some cases, it may be more cost-effective to shut down the old system and implement the new system all at once.

6. Once the new system is up and running for a while, it should be exhaustively evaluated. Maintenance must be kept up rigorously at all

times. Users of the system should be kept up-to-date concerning the latest modifications and procedures.

Phase 1: Project Identification and Selection

The first phase of SDLC is project identification and selection. In this phase, an organization's total information system needs are identified, analyzed, prioritized and arranged. Someone identifies the need for a new or enhanced system. Project identification and selection consists of three primary activities:

a. Identifying potential development projects
b. Classifying and ranking projects
c. Selecting projects for development

Phase 2: Project Initiation and Planning

In project initiation and planning phase a potential information systems project is explained and an argument for continuing or not continuing with the project is presented and a detailed plan is also developed for conducting the remaining phases of SDLC for proposed system. Two major activities are in this phase are investigation of the system problem or opportunity and the reasons why the system should or should not be developed by the organizations. The scope of the proposed system is determined in this phase.

Elements of project planning are:

- ✓ Describe the Project Scope, Alternatives and Feasibility
- ✓ Dividing the Project into Manageable Tasks
- ✓ Estimating Resources and Creating a Resource Plan
- ✓ Developing a Preliminary Schedule

- ✓ Developing a Communication Plan
- ✓ Determining Project Standards and Procedures
- ✓ Identifying and Assessing Risk
- ✓ Creating a Preliminary Budget
- ✓ Developing a Statement of Work
- ✓ Setting a Baseline Project Plan

Phase 3: Analysis

In analysis phase the current system is studied and alternative replacement systems are proposed. Analysis phase have several sub phases. The first is requirements determination. Methods for determining requirements are:

- ✓ Interviewing and Listening
- ✓ Directly observing Users
- ✓ Analyzing Procedures and other Documents
- ✓ Joint Application Design
- ✓ Using Prototype Identifying
- ✓ Processes to Reengineer

Then according to the requirement structure the system in context of their interrelationship and eliminate redundancies. Third, generate the alternative initial designs to match the requirements. Then from the alternatives determine the best solution that meet requirement properly and the cost, labor and technical levels of the organization is willing to commit to the development process. The output of the analysis phase is the alternative solution of the existing system.

Phase 4: Design

In the design phase description of the recommended solution is converted into logical and then physical system specifications. Design focuses on high level design like, what programs are needed and how are they going to interact, low-

level design (how the individual programs are going to work), interface design (what are the interfaces going to look like) and data design (what data will be required). During these phases, the software's overall structure is defined. Analysis and Design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. Much care is taken during this phase. The logical system of the product is developed in this phase.

Design phase has two parts: logical design and physical design. Logical design is the part of the design phase in which all functional features of the system chosen for development in analysis are described independently of any computer platform. Physical design is the logical specifications of the system from logical design are transformed into technology-specific details from which all programming and system construction can be accomplished.

Design phase highlighted on:

- ✓ Files and Databases
- ✓ Forms and Reports
- ✓ Dialogues and Interfaces
- ✓ System and Program Structure
- ✓ Distributed Systems

Phase 5: Implementation

In implementation phase information system is coded, tested, installed and supported the organization. In this phase the designs are translated into code. Computer programs are written using a conventional programming language or an application generator. Programming tools like Compilers, Interpreters, and Debuggers are used to generate the code. Different high level

programming languages like C, C++, Pascal, Java are used for coding. With respect to the type of application, the right programming language is chosen.

Implementation activities also include initial use support such as finalization of documentation, training programs and ongoing user assistance. Elements of implementation phase are:

- ✓ Coding
- ✓ Testing
- ✓ Installation
- ✓ Documentation
- ✓ Training
- ✓ Support

Phase 6: Maintenance

In the maintenance phase information system is systematically repaired and improved. After implementation users faces some problems about how the system works and often think of better ways to perform its function. So modification or change is needed. Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period. Major activities that occur in maintenance phase are:

- ✓ Obtaining Maintenance Requests
- ✓ Transforming Requests into Changes
- ✓ Design Changes
- ✓ Implementing changes

There are several types of maintenance. Corrective maintenance refers to changes made to repair defects in design, coding and implementation of the system. Adaptive maintenance involves making changes to an information

system to evolve its functionality to changing business needs or technologies. Perfective maintenance involves making enhancements to improve processing performance, interface usability, and add desire requirements and features. Preventive maintenance changes the system to reduce the chance of future system failure.

1.3 Identifying and Gathering Requirements

Defining user requirements requires an understanding of how the system works and what its problems are. There are many important issues to consider in getting a clear picture of a system. One is to look at the current business processes in the system and identify the tasks in those processes. Or one can begin by examining particular system functions and their tasks. The tasks can then be examined in detail. Such examination can identify the users, who carry out the tasks, the interactions between the users, the tools they use and the artifacts on which they operate. Thus an analyst must always consider the users and what they do. To have preconceived ideas about a system and analysts must approach any study with an open mind. There are two main ways of doing this – namely by:

- ✓ Asking questions by interviewing people in the system, through surveys and questionnaires, or by electronic means using e-mail or a discussion database.

- ✓ Directly observing the users.

1.3.1 Gathering Information by Interviewing

Interviewing is perhaps the most commonly used technique in analysis. There is no real way to avoid interviews, as they must precede any other method for gathering information about system requirements it is always necessary first

to approach someone and ask them what their problems and priorities are, and later to discuss with them the results of the analysis.

Questionnaire: Gathering information by filling in a form. It provides an alternative to interviews for finding out information about a system. Questionnaires are made up of questions about information sought by the analyst. The questionnaire is then sent to the user, and the analyst analyzes replies.

Electronic data gathering: Electronic communication systems are increasing being used to gather information. Thus it is possible to use electronic mail to broadcast a question to a number of users in an organization to obtain their viewpoint on a particular issue.

In the organization some time must be spent in talking to or interviewing people. Such interviews may only be in the initial stages to identify the major issues, or they can proceed throughout the entire life of a project. Interviewing is the main approach used to analyze large structured system. Interviewing is gathering information by asking questions.

It is important to establish a good relationship with interviewees and this should start right at the beginning.

There are some basic premises that you should always be aware of when conducting interviews. First, you must gain the confidence of your interviewees. To do this you must convince the interview of your own abilities and show that you are proceeding in an organized way and will not waste their time. You must also be sympathetic to their problems and not become aggressive and create the impression that you are there to apportion blame.

It is not important to force solutions upon users but rather to play the role of an advisor. Computer jargon should not be used to impress the user, though interviewers should explain the limitations of the computer in user terms and describe how it can assist users in their work. You should not try to elicit the response you want by asking leading questions.

Interviewers should also take care to ensure that they obtain all the needed information from interviews. It helps to let the user to know what information is required from the interview. The interviewer should then seek this information gradually and be precise and direct in their questioning.

The model and information about it grows at each interview. Analysts will be satisfied because their knowledge about the system is continually growing. The interviewees will more cooperative because they can see that their time is being used productively.

Interview search procedures: Search procedure is the process followed to gather information about a system. Information gathering in large systems with many information sources must proceed in an organized way to ensure that all the relevant information needed to build the system is obtained. Analysts must determine what information needs to be gathered and the users who are to supply this information. They must then seek this information in an orderly way so that vital information is not neglected and being asked the same questions does not repeatedly bother people. To avoid such problems, it is necessary to develop a search procedure for gathering information. This procedure will define the steps to be followed in gathering information and the information to be obtained at each step. Such steps usually require the search to proceed in a top-down manner, objectives to be set for each step, and an appropriate search method to be chosen for each step.

The search procedure suggests what order is to use to search information sources and what methods are to be used as the search proceeds. Thus the search procedure becomes a plan starting what information is to be obtained from each source and what sequence is to be used to search the sources.

The interview plan specifies:

- ✓ The users to be interviewed
- ✓ The sequence in which the users are interviewed and
- ✓ The interview plan for each user

The first step in developing an interview plan is to identify the users to be interviewed. Often an organization chart can be used to identify such users. This chart describes the organization's units, the positions in these units and each position's occupant. The analyst uses the project's terms of reference to select the organizational units that fall within the boundary of the system study and is likely to be affected by any new system. Persons in these units then become candidates for interviewing. It is usually wise to begin interviewing at the top levels of the organizational areas, in order to get support and cooperation from managements before beginning to look into particular organizational activities or suggesting new solutions. Management may then often suggest other users that should be interviewed and is more likely to support any proposed changes.

There are also some common goals in each interview. Preparation for the interview is always essential. The analyst should have an idea of what information is needed from the interview and ask direct questions to get this information. If the current interviewee cannot answer, the analyst should ask for advice about where to go next.

Preparation for the interview is always essential. The analyst should have an idea of what information is needed from the interview and ask direct questions to get this information.

One of the most important points about interviewing is what question you need to ask. It is important to ask the right questions in the correct order to get the most out of interviews. Questions can be characterized by their subject content and type. Obviously, the subject content of a question will depend on the specific system study. However, the type of question can be generalized and there are general guidelines to help you to choose the most appropriate questioning method for the interview in question.

It is often convenient to make a distinction between three kinds of questions. These are

- ✓ Open questions
- ✓ Closed questions
- ✓ Probes

The above three are explain in the following way:

Open Questions: Open questions are general questions that establish a person's viewpoint on a particular subject. It requires the responder to express a viewpoint.

The first interview with a user should emphasize open questions. This enables analysts to identify the values a user places on the system helps in gaining impression about the users like and dislike. This may then shape future

interviews. Open questions also prompt the user to volunteer more detailed information so that closed questions may be unnecessary.

Closed Questions: Closed questions are more specific. It is a question that requires a direct answer. A closed question often restricts an interviewee to some specific answer. This may be a number, an explanation of a report, or a reason for doing something. The answer to closed questions can then be followed with a probe to get more detail.

It is possible to start with open questions follow then closed questions and then finished up with some more open questions. This approach is used mostly in follow-up interviews. It is an easy way to gradually lead a user into detailed questioning. Some people, when answering an open question, may provide enough information to make future detailed questions unnecessary. It is also possible to start with detailed questions and finished with open questions.

Probe: Probes are questions that follow up an earlier answer.

Interviewers must often select a particular questioning method for an interview. This method will use alternate sequence of closed and open questions.

1.3.2 Gathering Information by Observation

Interviewing and other ways of asking questions are characterized by having analysts learn about the system without themselves getting involved in the system. Interviews often emphasize what an individual does and how to support them, rather than looking at the individual's relationship to their group. The analyst then has to correlate findings about individuals and determine how

the whole group works. It is also the interviewer's responsibility to find out the objectives of each individual and try to make them cohesive.

Using ethnography: Ethnography is gathering information by observation. It is not a new field but, rather, a new approach to analyzing computer system requirements. One of its most important goals is not to superimposing the interviewer or analyst's viewpoint on the system but to use the viewpoint of the people within the system.

Analysis by observation: The goal here is to observe what people do in an unobtrusive way. The best way to do this is by video recording. It is important in video recording to ensure that the presence of the video camera itself does not alter behavior while at the same time collecting sufficient in depth information to make useful observations.

From observation to design: Another characteristic of the ethnographic approach is that objectives are developed with user participation. Considerable time, for example, may be spent on the design of workspaces through experimentation. Such workspaces not only include the computer screen but also the physical space used for any manual work on associated documentation or paperwork.

CHAPTER II

PRESENT SYSTEM ANALYSIS

2.1 Overview of BRAC

BRAC, a national private development organization, set up in 1972 by Mr. Fazle Hasan Abed, began as a relief organization focused on resettling the refugees returning from India after the War of Independence in 1971. This task over, BRAC redirected its focus to the issue of poverty alleviation and empowerment of the poor, especially women, in Bangladesh's rural areas. BRAC, the acronym for Bangladesh Rural Advancement Committee, has become its identity and it stands for working for the poor and the marginalized. From its modest birth in 1972, it is now one of the world's largest national NGOs, diverse in its operations with over 28,000 regular staff and 34,000 part-time teachers, working in 61,924 villages in all 64 districts of Bangladesh. BRAC has progressed by learning from experience and through a responsive and inductive process. Adjusting its strategy to prevailing circumstances, it does not pursue any rigid development model. From the mid-70s to early 80s, our holistic and flexible approach to rural development became the internationally accepted standard. It has been called upon to assist countries in Africa, Asia, and most recently in war ravaged Afghanistan.

BRAC diagnoses poverty in human terms. Women with social, cultural, technological, and structural constraints have been able to transform themselves as contributors not only to their families' well-being, but also to national production and development by increasing their access to economic and social

resources with BRAC's assistance. Currently, BRAC promotes income generation and social development of the poor, mostly landless rural people of Bangladesh through micro credit, health, education, and training programs.

BRAC Economic Development Program has so far organized over 4.07 million poor landless people, mostly women, into 119,836 Village Organizations (VOs), each having 30-40 members. These groups serve as forums where the poor can collectively address the principal structural impediments to their development, receive credit, and open savings accounts. While BRAC believes that micro credit is an important tool in breaking the cycle of poverty, it also places equal emphasis on training of its members in income generating activities and in facilitating their linkages with consumer markets. BRAC's credit program, initiated in 1976 has disbursed Taka 107,310 million, with a recovery rate at 98.04%. The savings deposited with BRAC now stand at Taka 6,285 million. The average loan size is Taka 6,879, where no collateral is required. BRAC organizes a number of social development initiatives designed to increase members' awareness of their rights and responsibilities, and to facilitate addressing issues of discrimination in their villages. BRAC continuously challenges itself to question its own assumptions, implicit and explicit, and reviews them in the light of unfolding reality and experiences. Since January 2002, BRAC Economic Development Program has been implementing its newest project, challenging the Frontiers of Poverty Reduction Targeting the Ultra Poor (CFPR-TUP). This program uses a specific set of criteria to identify those women who are in the very margins of society, and are too poor to take advantage of standard micro credit options. After identifying these ultra poor women, BRAC gives them income-earning assets, provides training in the women's chosen income generating activity, and offers education and health care services. Thus the ultra poor are nurtured to eventually merge into mainstream micro credit programs. BRAC's ultra poor program has already garnered international attention and is setting the standard for other development organizations to emulate. BRAC's Health, Nutrition and Population Program takes a broad approach to the health needs of the poor by providing basic curative and preventive services to more

than 31 million people. Trained health workers and female health volunteers work to raise awareness among the rural poor of health issues that directly impact their lives and families. It seeks to reduce maternal and child mortality, and reduce vulnerability to common diseases. Services are offered to control infectious diseases such as tuberculosis, acute respiratory infections, diarrhea, etc. The program also provides services to pregnant women that improve their health and nutrition status. The reproductive health needs of the community in general, with particular focus on BRAC members, are addressed through education on family life, contraception, STD/RTI control, and awareness of HIV/AIDS. BRAC encourages rural people to use safe water and practice hygienic sanitation. The Nutrition Facilitation Program is working as a partner of the Bangladesh Government's national nutrition initiative.

BRAC works with people whose lives are dominated by extreme poverty, illiteracy, disease and other handicaps. With multifaceted development interventions, BRAC strives to bring about positive change in the quality of life of the poor people of Bangladesh. BRAC firmly believes and is actively involved in promoting human rights, dignity and gender equity through poor people's social, economic, political and human capacity building. Although the emphasis of BRAC's work is at the individual level, sustaining the work of the organization depends on an environment that permits the poor to break out of the cycle of poverty and hopelessness. To this end, BRAC endeavors to bring about change at the level of national and global policy on poverty reduction and social progress. BRAC is committed to making its programmers socially, financially and environmentally sustainable, using new methods and improved technologies. As a part of its support to the program participants and its financial sustainability, BRAC is also involved in various income generating enterprises.

Poverty reduction programs undertaken so far have bypassed many of the poorest. In this context one of BRAC's main focuses is the ultra poor. Given that development is a complex process requiring a strong dedication to learning,

sharing of knowledge and being responsive to the needs of the poor, BRAC places a strong emphasis on their organizational development, simultaneously engaging itself in the process of capacity building on a national scale to accelerate societal emancipation. The fulfillment of BRAC's mission requires the contribution of competent professionals committed to the goals and values of BRAC. BRAC, therefore, fosters the development of the human potential of the members of the organization and those they serve. In order to achieve its goal, wherever necessary, BRAC welcomes partnerships with the community, like-minded organizations, governmental institutions, and the private sector and development partners both at home and abroad.

2.1.1 Micro credit Program

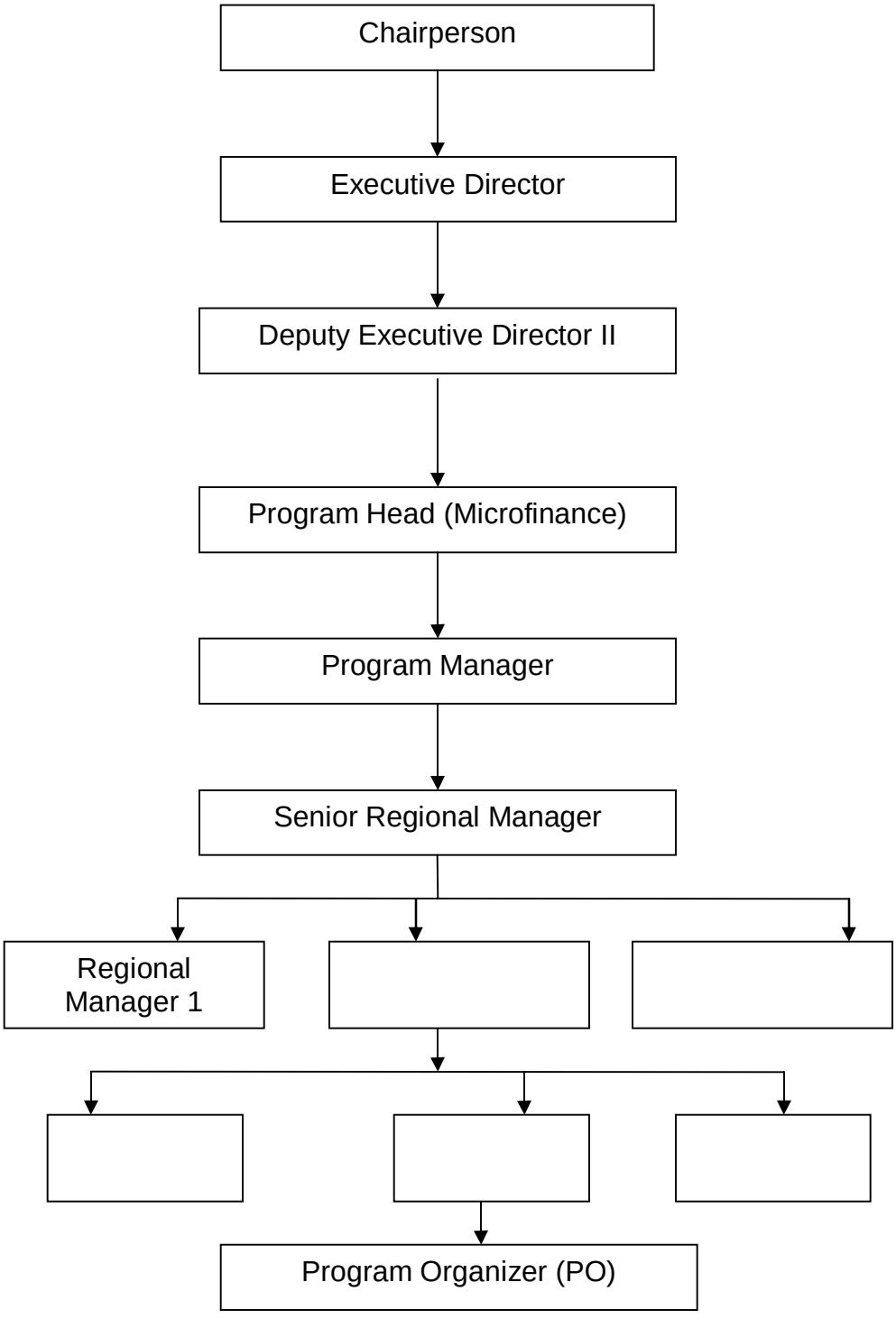
Bangladesh is characterized by high levels of poverty accompanied by low productivity. Because the poor cannot provide collateral and typically deal in small denominations of money, they are denied access to the formal banking system, and are thus deprived of the facilities to borrow, save and invest in productive activities. In addition, moneylenders from the informal credit markets charge very high interest rates restricting poor people's access to credit earnings. Making credit universally available is therefore an essential strategy in reducing income poverty.

Launched in 1974, BRAC's Micro credit Program aims to:

- Make credit available to poor women, especially in rural areas.
- Provide credit at a reasonable price.
- Involve poor women in income generating activities through credit provision.
- Promote the economic development of the country by increasing the income level of the rural poor.
- Operate self-sustaining credit activities.

The micro credit program of BRAC is a tool for poverty alleviation and empowering the poor. Lack of access to the formal banking system deprives them of the facilities to borrow, save and invest in productive activities and this is a major reason why poor people remain poor. The formal banking sector also requires collateral. Making credit available to the rural poor enables them to become involved in different income generating activities which, in turns, allows them to become economically self-reliant. Through this process BRAC's micro credit program works to create a self-sustaining and reliable financial service program for the poor.

2.1.1.1 Hierarchy of the existing system


Chairperson
Executive Director
Deputy Executive Director II
Program Head (Microfinance)
Program Manager
Senior Regional Manager
Regional Manager 1
Program Organizer (PO)

2.2 Credit

Credit is provided to its VO members to initiate different income generating activities. While loans for individual and joint activities do not require collateral, members must have some savings with BRAC before they are eligible for loans. Credit operations are carried out through a Revolving Loan Fund (RLF). This RLF consists of donors' fund, members' savings, Polli Karmo Shohayok Foundation (PKSF) loan and other loans. Loans realized from VO members are credited to and form part of the RLF for extending further credit. A 2% loan loss reserve is kept to cover the risk of bad debts and death. Regular borrowing and payments allow the borrower to take larger loans.(Ref: Appendix-A)

Key Features of BRAC Micro Credit Loan

- Stanchion Loan range: Tk.1,000 to Tk. 20,000
- Service Charge: 15%
- Repayment mechanism: Equal weekly installments
- Loan products: General loan, sector program loan, housing loan and emergency loans given at the time of disaster

In 2002, Tk. 1,707 crore has been disbursed to 2.9 million borrowers with repayment rate of over 98%.

2.3 Savings

Savings is an important component of microfinance services. Experiences show that there is a positive correlation between savings and sustainable credit operation. From the member's point of view, savings represent an opportunity to save in small amounts to form a lump sum that earns interest. This is an opportunity that the formal market or regular financial institutions do not provide.

A VO member may save in three ways:

a. Weekly Personal Savings: On average, members are required to save a minimum Tk. 5 every week.

b. Compulsory Savings: When VO members take loans, it is mandatory that they deposit 5% of the loan amount into their savings account. A member can withdraw 50% of their net savings at any time of the year.

c. Current Account Savings: Current Account Savings has been recently introduced that bear no interest but allow the group members to make unlimited withdrawals. This has the following features:

Convenience: The Area Office is conveniently located to allow all members to deposit money into their current accounts.

Liquidity: Members can withdraw any amount of savings from these accounts at any time.

Safe Keeping: The Area Office is a safe place to keep savings instead of their houses.

By the end of 2002, the total savings stood Tk. 498 crore.

2.4 Death Benefit

A death benefit policy has been introduced for its VO members since June 1990.

The key features of BRAC's death benefit are:

- All VO members irrespective of borrowers are entitled to this benefit.
- Death benefit service provides Tk 5,000 to the dependants of the deceased member.
- No premium is charged to the members. The cash benefits are paid out of the service charge earned through BRAC credit program.
- Outstanding loans of the deceased are written off.

2.5 Information Flow of Existing System

Regional Office

Send data to the

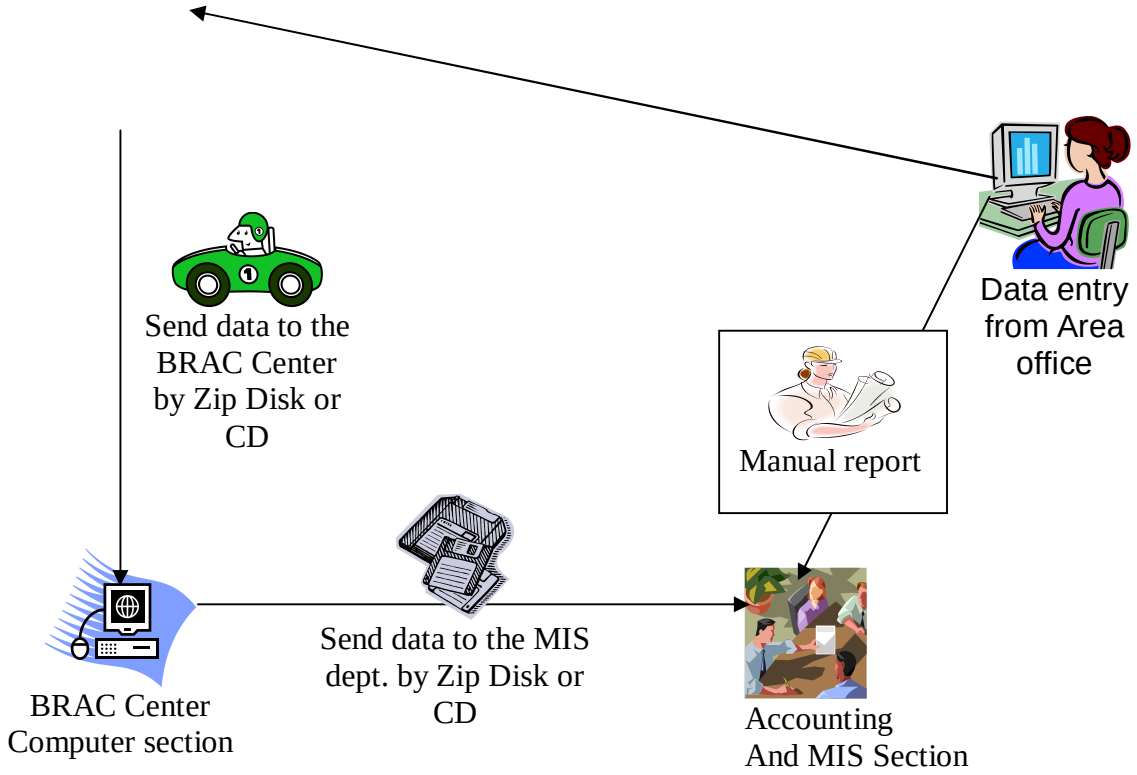


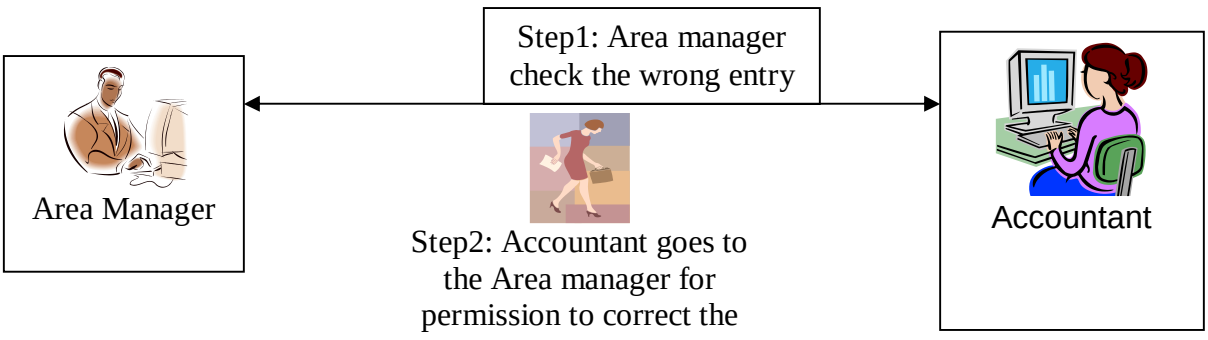
Fig.2.1 Data flow form Area office to BRAC centre computer section.

When one new member start saving then all kinds of information about the new member posted in to the computer by accountant. PO collect installment, saving and other information by manual sheet. Then all information goes to the area office accountant. Accountant post those data into the computer. Accountant gives give money as loan according to the area manager permission

and post it into the computer. Last week of the every month accountant export all information by CD or Zip disk to the regional office accountant. And also create some manual report for BRAC center computer section and MIS department and accounting section, which send physically in every weekly and also monthly. As document one CD or Zip disk keep also area manager. Then regional office sends this CD or Zip disk to the BRAC center computer section by physically. When BRAC center computer section received all data form the whole country they post it into the computer section database. And send some data by CD or Zip disk to the MIS and BRAC center main accountant office according to their need. Then MIS and accounting department create their necessary report. And also some time they send request over phone to the computer section for some important reports. Then computer section cerate those report and send its to the this department.

Area office, regional office and the BRAC center computer section exchange their others information over phone, fax and mobile phone which required more time and also expensive.

2.6 Flow of the Existing System for Correcting "Wrong-Data"



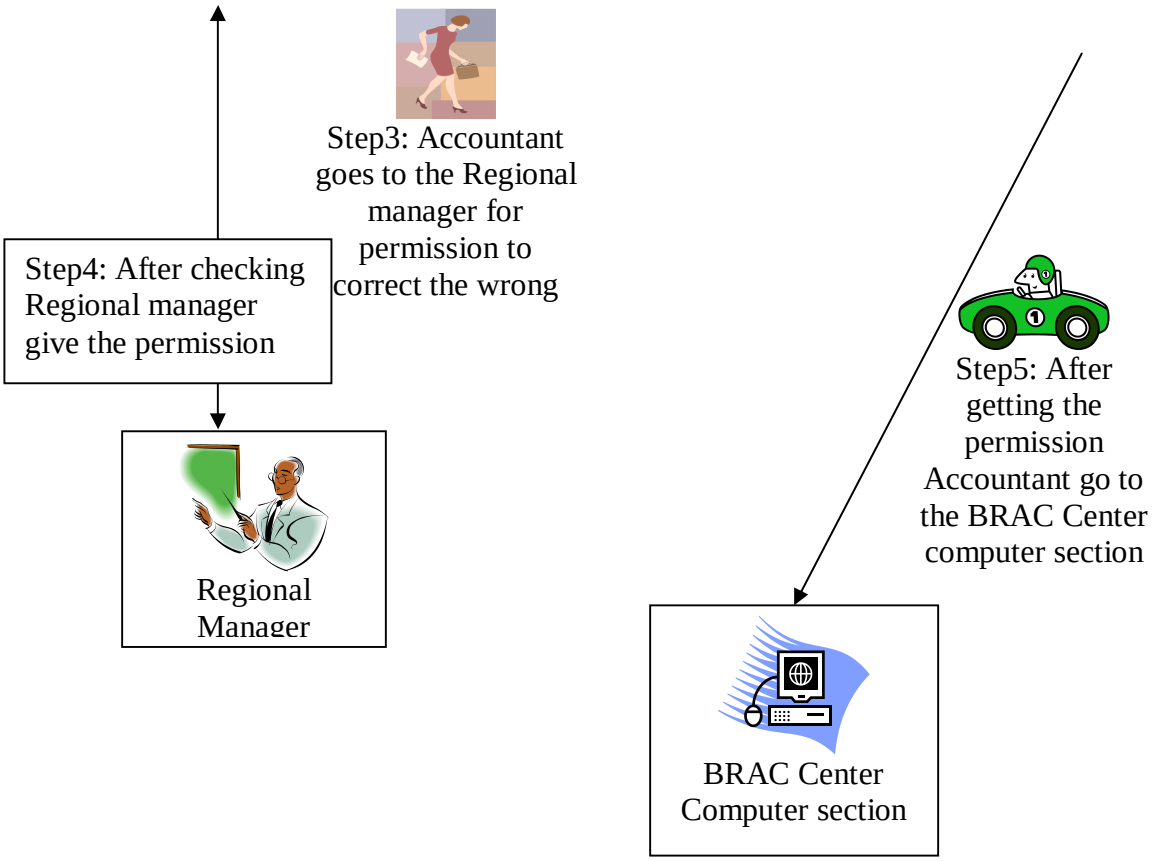


Fig2.2. flow for wrong data correction from Area office.

Area accountant give the entry for all kinds of data. If s/he made any mistake to give entry then it is very difficult way to correct it. For correcting "wrong-data" the following steps are followed by accountant:

Step1: Area manager check the wrong entry.

Step2: Accountant goes to the Area manager for permission to correct the wrong entry.

Step3: Accountant goes to the Regional manager for permission to correct the wrong.

Step4: After checking Regional manager give the permission

Step5: After getting the permission Accountant go to the BRAC Center computer section to correct the.

And then BRAC center computer section operator will correct those wrong entries.

This above process is very difficult and also expensive.

2.7 Dissection of the Overall System:

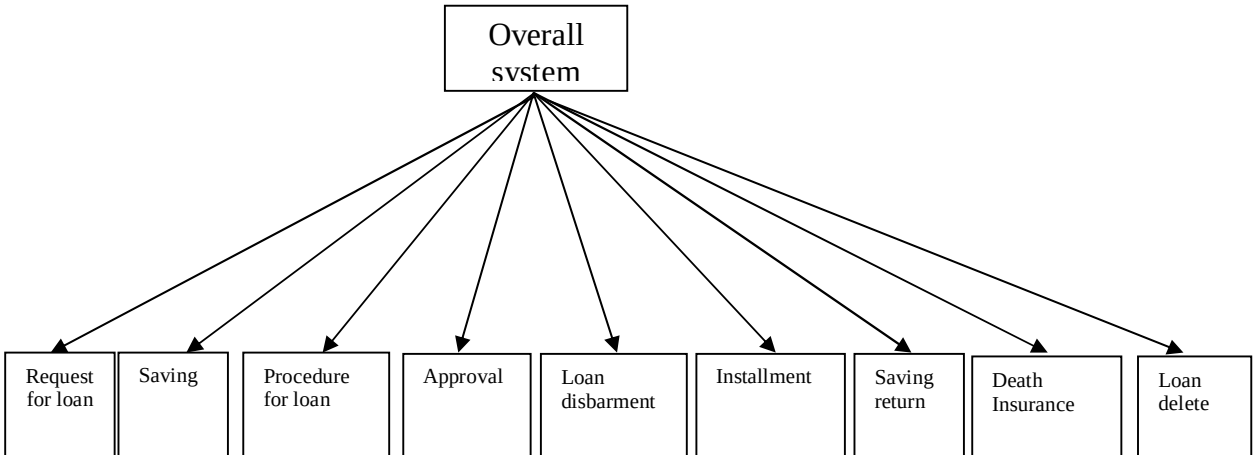


Fig. 2.3 Dissection of the overall system

2.7.1 Request for loan

When anyone wants loan from BRAC then he have to go to the PO or area manager of his area or an existing group whose are already taken loan.

If he goes to the PO

1. The PO personally judge the man's economic condition and also ask others of the villagers about him. If every thing is correct then
2. PO gives advice him to join a group.
3. Again asking for oral vote then other group members selected him by oral vote.

If the new member directly goes to the area officer then area manager assigns him to his area's PO then the above process apply.

If he goes to any existing group to join for loan, then in the meeting existing group member propose for taking the new member in their group and group members select him by oral vote.

2.7.2 Saving

As for taking loan one member has to save certain amount of money, so savings are important issue for getting loan.

For start saving her/him must fulfill this following condition:

1. S/he must have under a group.

For loan s/he has to continue saving for six months then he can propose for loan. Loan amount depends upon how much he saves in six months.

2.7.3 Procedure for Loan

While giving loan to a new member the earning sources of that member, economic condition are checked by PO and sometimes by area manager. Here mainly checked is if he gets loan whether he can give back the loan installment. PO and sometimes area manager have to go to the field and ask the neighbor of him about his economical and family background and also about him although s/he has under a group.

PO and Area manager also check that

1. S/he has under a group or not.
2. Is his/her saving enough for loan?
3. Then Area manager accept the request foe loan.

For loan s/he has to continue saving for six months then he can propose for loan. Loan amount depends upon how much he saves in six months.

2.7.4 Approval

When request for loan is accept then a member propose for a certain amount of loan. Area manager approve the loan. If area manager thinks that loan amount exceeds limits then he can reduce the amount and approve that much amount of money as loan.

2.7.5 Loan disbursement

For loan disbursement follow these following steps

1. Accountant get one loan approved copy from Area manager.
2. Member come with loan approved paper to the accountant for money.
3. Accountant cross check both loan approved paper which are found from Area manager and member.
4. If everything is OK then accountant gives that amount of money to the member as loan.

2.7.6 Loan installment

After getting the loan money after the very first week that member have to give back his first installment amount. Because of installment must be paid in every week. Loan is paid within 46 weeks. So total loan amount is divided with total number of weeks and interests are also calculated here. With this loan installment member also paid saving which is the five percent of the loan amount.

2.7.7 Saving return

If any member wants his savings back then he has to inform it to PO, then PO inform it to the area manager. Then area manager ask for corresponding data of that member from accountant. The main criteria of savings return is the amount of loan must less than the savings. If these criteria meet then area manager informs this to the regional manager and regional approves amount of savings return.

2.7.8 Death insurance

When one existing member died then his or her nominee will be allowed to get the death insurance. Before getting death insurance Area manager will ensure two things:

1. The death person's must have authorized nominee.
2. His or her loan installment has been paid properly or not.
3. Is membership renewed yearly basis or not.

After getting the permission from Area manager the nominated person can get this death insurance.

2.7.9 Delete loan

If any member does not want to continue the loan account any more then he inform it to PO. The PO informs it to area manager and area manager take the information about that member from accountant. Then after balancing the loan and savings when the entire amount becomes zero then member is deleted.

CHAPTER III. INTRODUCTION OF DIFFERENT UML

3.1 What is UML?

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is very important parts of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

3.2 Goal of the UML

The primary goals in the design of the UML are:

- Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
- Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

As the strategic value of software increases for many companies, the industry looks for techniques to automate the production of software and to improve quality and reduce cost and time-to-market. These techniques include component technology, visual programming, patterns and frameworks. Businesses also seek techniques to manage the complexity of systems as they increase in scope and scale. In particular, they recognize the need to solve recurring architectural problems, such as physical distribution, concurrency, replication, security, load balancing and fault tolerance. Additionally, the development for the World Wide Web, while making some things simpler, has exacerbated these architectural problems. The Unified Modeling Language (UML) was designed to respond to these needs.

3.3 The Use Case Model

The use case model includes the actors, the system, and the use cases themselves. The set of functionality of a given system is determined through the study of the functional requirements of each actor, expressed in the use cases in the form of 'families' of interactions. Actors are represented by little stick people who trigger the use cases, which are represented as ellipses contained within the system.

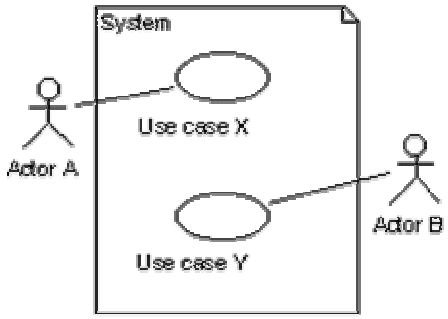


Fig.3.1: Example of Use Case Model

An actor represents a role played by a person or a thing that interacts with a system. Actors are determined by observing the direct users of a system – those who are responsible for its use or its maintenance – as well as the other systems that interact with the one under study. The same physical person may play the role of several actors (vendor, client). Additionally, several people may all play the same role and therefore act as the same actor (all the customers). The name of the actor describes the role played by the user.

.

There exist four main categories of actors:

Principal actors – people who use the main system functions. In the case of a cash dispenser, they are the customers.

Secondary actors – people that perform administration or maintenance tasks. In the case of a cash dispenser, it is the person in charge of reloading the money cassette contained within the dispenser.

External hardware – the unavoidable hardware devices that are part of the application domain and must be used. It has nothing to do with the computer on which the application is executed, but pertains to the other hardware peripherals. In the case of a cash dispenser, this may correspond to the printer.
Other systems – the other systems with which the system must interact. In the case of a cash dispenser, the bank network system that manages a set of dispensers plays the role of an actor.

Once identified, actors must be described clearly and precisely, in three or four lines at most. When there are many actors within a system, it is advisable to group them in categories to facilitate navigation within the use case model.
Use cases are determined by observing and specifying, actor by actor, the interaction sequences (scenarios) from the user's standpoint. They are described in terms of the information exchanged and the way the system is used. A use case groups a family of usage scenarios according to a functional criterion. Use cases are abstractions of dialog between the actors and the system: they describe potential interactions without going into the details of each scenario.
Use cases must be seen as classes whose instances are the scenarios. Each time an actor interacts with the system, the use case instantiates a scenario. This scenario corresponds to the message flows exchanged by objects during the particular interaction that corresponds to the scenario. Analysis of requirements by use cases is very well complemented by an iterative and incremental approach.

The scope of use cases goes far beyond solely defining of the requirements of the system. Indeed, use cases come into play throughout the lifecycle, from the specification stage to system testing, through the analysis, design, implementation, and documentation stages. From that standpoint, it is possible to navigate first towards the classes and objects that collaborate to satisfy a

requirement, then towards the tests that verify the system performs its duties correctly.

3.4 Class Diagram:

Class diagrams express, in a general way, the static structure of a system, in terms of classes and relationships between those classes. Just as a class describes a set of objects, an association describes a set of links; objects are class instances, and links are association instances. A class diagram does not express anything specific about the links of a given object, but it describes, in an abstract way, the potential links from an object to other objects.

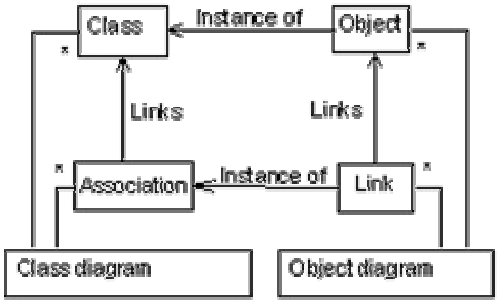


Fig.3.2: Example of Class Diagram Model

Simplified metamodel showing class diagrams to represent classes and relationships.

3.5 Interaction Diagrams:

An interaction expresses the behavior resulting from the collaboration of a group of instances. An interaction may be displayed with respect to the time perspective (by sequence diagrams), or with respect to the space perspective (by collaboration diagrams).

Interactions comprise the following main elements:

- Instances: which are concrete examples of a type
- Links: which connect instances and are used to support message broadcasts
- Messages: which trigger the operations
- Roles: played by the ends of the links

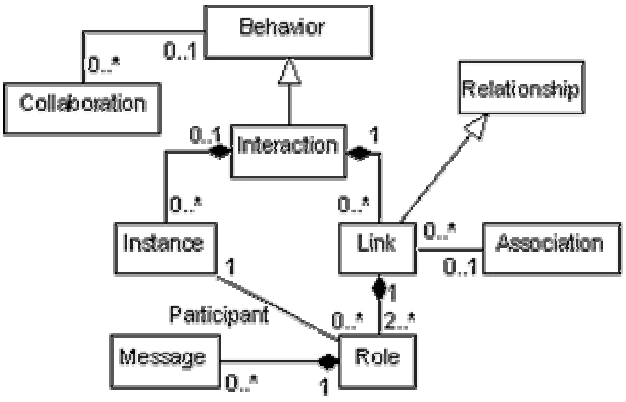


Fig.3.3 Example of Simplified metamodel of Interactions

3.5.1 Collaboration diagram:

Collaboration is a mechanism composed of structural and behavioral elements. Collaborations provide an organization mechanism, but unlike packages they have an identity and a semantic scope. A given element may play a role in several collaborations.

Collaboration includes two types of construction: a context, which is composed of a description of the static structure of the objects being considered, and an interaction represented by a sequence of messages exchanged by these objects. The two viewpoints are necessary to document the behavior completely, but each viewpoint may be displayed independently.

Collaborations are used, depending on their level of detail, to describe specifications and express implementations. The table below summarizes the model elements that may be described by a collaboration:

Table3.1: summarizes the model elements that may be described by collaboration:

Specification	Type	Operation	Use case
Implementation	Class	Method	Use case implementation

Collaborations also exist in a generic form (model), parameterized by classes, associations, attributes, and operations. A generic collaboration is called a pattern, or a scheme. Patterns always have a name, in contrast to collaborations that may remain anonymous. A Simplified metamodel representation of collaborations.

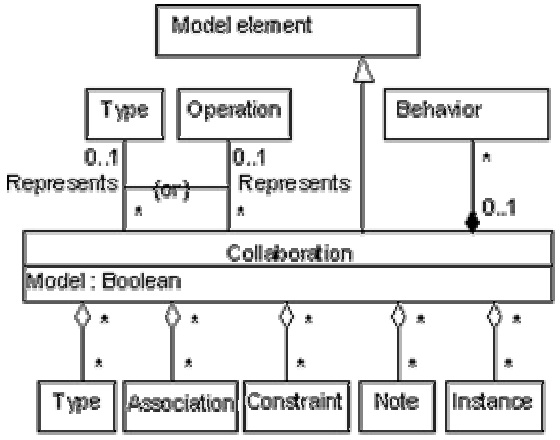


Fig.3.4 Example of collaboration diagram.

3.5.2 Sequence diagram:

A system sequence diagram is a fast and easily created artifact that illustrates input and output events related to the system under discussion. The UML contains the notation in the form of sequence diagrams to illustrate events from external actors to a system.

Before proceeding to a logical design of how a software application will work, it is useful to investigate and define its behavior as a “Black box.” System behavior is a description of what a system does, without explaining how it does it. One part of the description is a sequence diagram. Other parts include the use cases, and system contracts.

Use cases describe how external actors interact with the software system. It is desirable to isolate and illustrate the operations that an external actors request of a system, because they are an important part of understanding system behavior. The system sequence diagrams as a notation that can illustrate actor’s interactions and the operations initiated by them.

Finally a system sequence diagram (SSD) is a picture that shows, for a particular scenario of a use case, the event that external actors generate, their order, and inter-system events. All systems are treated as a black box; the emphasis of the diagram is events that cross the system boundary from actors to systems. An SSD usually should be done for the main success scenario of the use case, and frequent or complex alternative scenarios.

Sequence diagrams demonstrate the behavior of objects in a use case by describing the objects and the messages they pass. The diagrams are read left

to right and descending. The example below shows an object of class 1 start the behavior by sending a message to an object of class 2. Messages pass between the different objects until the object of class 1 receives the final message.

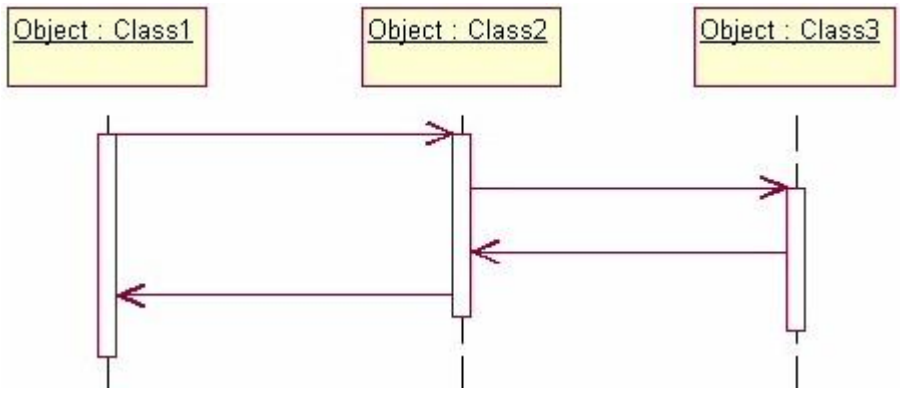


Fig.3.5: Example of sequence Diagram that pass messages

3.6 Package Diagram:

Packages provide a general mechanism for partitioning models and grouping model elements. Each package is represented graphically by a folder.

Fig3.6: Example of a package.

Packages divide and organize models in much the same way that directories organize file systems. Each package corresponds to a subset of a model and contains, depending on the model, classes, objects, relationships, components, or nodes, as well as their associated diagrams.

Decomposition into packages is not the basis for a functional decomposition; each package is a grouping of elements according to purely logical criteria. The general system layout (the system architecture) is expressed by the hierarchy of packages and by the network of dependency relationships between packages.
The stereotypes »category » and »subsystem » allow distinguishing between packages of the logical view and packages of the implementation view as required.

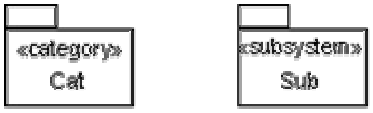


Fig.3.7: Example of package and subsystem.

A package defines a namespace, so that two distinct elements contained in two distinct packages may have the same name. A package may contain other packages, with no limitation on nesting levels. A given level may contain a mixture of packages and other model elements, in the same way as a directory may contain other directories and files.

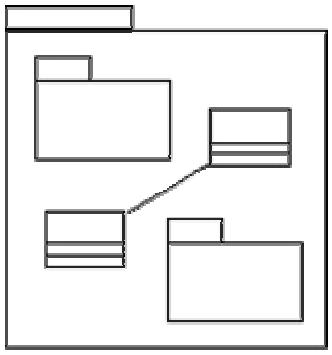


Fig.3.8 Example of Package diagram

Every element belongs to a package. The package located at the highest level is the root package of the whole model. A class contained within one package may

also appear in another package in the form of an imported element, by means of an inter-package dependency relationship.

3.7 State diagram:

Each object is in a particular state at a given point in time. States are represented as rounded rectangles; each state has an identifying name.

Fig.3.9 Example of Simple State

States are represented using rounded rectangles – each state has a name that must be unique within a given lexical scope.

States are characterized by the concepts of duration and stability. An object is not always in the same state at a given time, and an object cannot be in an unknown or undefined state. A state is the image of an instantaneous combination of the values contained in the object's attributes, and the presence or the absence of links from the given object to other objects.

The state machines defined by UML are deterministic. Therefore, a state chart diagram must not leave any room for ambiguous constructs. This means, in particular, that it is always necessary to describe the system's initial state. For a given hierarchical level, there is always one and only one initial state. Conversely, it is always possible to have several final states that each

correspond to a different end condition. It is also possible to not have any final state – in the case of a system that never stops, for example. The initial state is represented by a big black dot. A final state is represented by a big black dot surrounded by a circle.

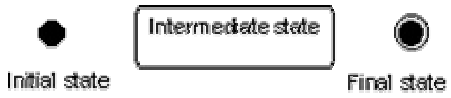


Fig.3.10: Example of Different State

3.8 Activity Diagram:

Activity diagrams describe the workflow behavior of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel.

Activity diagrams should be used in conjunction with other modeling techniques such as interaction diagrams and state diagrams. The main reason to use activity diagrams is to model the workflow behind the system being designed. Activity Diagrams are also useful for: analyzing a use case by describing what actions needs to take place and when they should occur; describing a complicated sequential algorithm; and modeling applications with parallel processes.

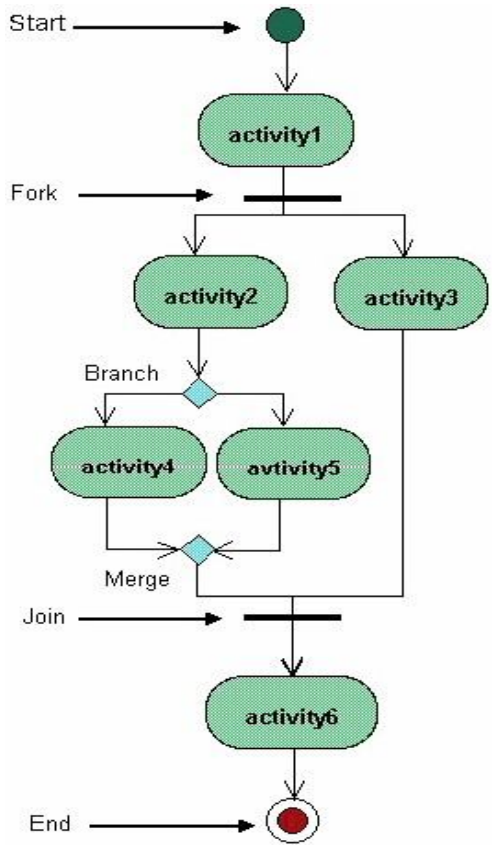


Fig.3.11 Example of activity diagram

Diagrams are read from top to bottom and have branches and forks to describe conditions and parallel activities. A fork is used when multiple activities are occurring at the same time. The diagram below shows a fork after activity1. This indicates that both activity2 and activity3 are occurring at the same time. After activity2 there is a branch. The branch describes what activities will take place based on a set of conditions. All branches at some point are followed by a merge to indicate the end of the conditional behavior started by that branch. After the merge all of the parallel activities must be combined by a join before transitioning into the final activity state.

CHAPTER IV. PRESENT SYSTEM ANALYSIS BY DIFFERENT UML

4.1 Overall System:

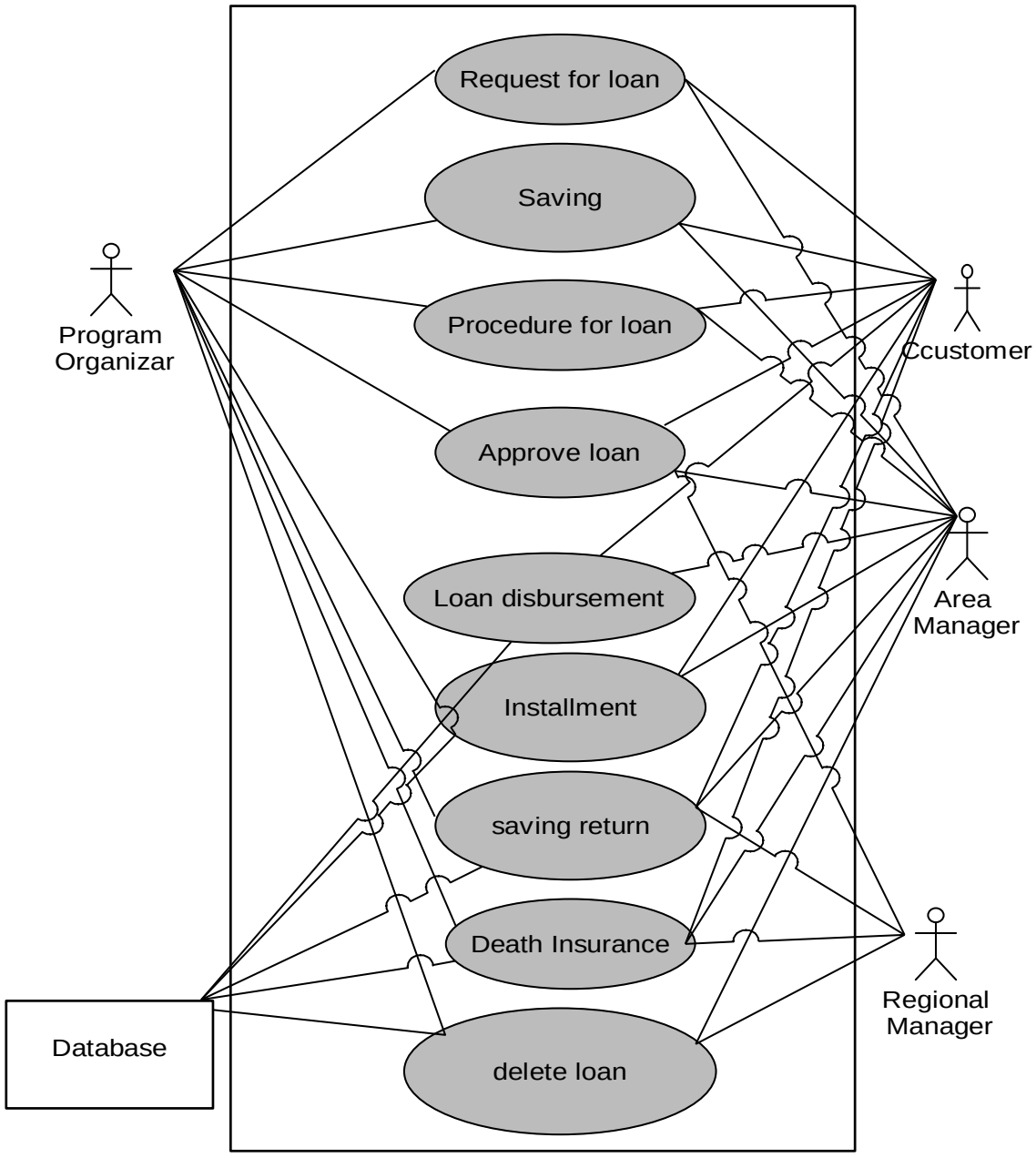


Fig. 4.1 Use case for BRAC micro credit (Ref : 2.5)

Use Case: Request for loan

Primary Actor: Customer, Program Organizer, Area manager.

Stakeholders and interest:

- Customer: Want registration for loan and saving.
- Program Organizer: Want to get the information about the requested member and take oral selection for membership confirmation.
- Area manager: Want to get the information about the requested member, which are required for membership.

Preconditions: There should be applying with group.

Success Guaranteed (Post condition): Membership completed and database updated successfully.

Main Success Scenario (or Basic Flow):

1. Customer request to P.O for membership.
2. Program Organizer takes oral selection from the Group.
3. If yes then this request ready to go for area manager.
4. Then area manager got the request and check all requirement for membership.
5. If yes then request accept.

Extension s (or Alternative Flows):

1a. Customer said to cancel loan request.

1b. Group Request to P.O for loan request.

1c. P.O cancel the loan request.

2a. Oral selection can be cancel for Absence of some group members.

2b. Program organizer start oral selection from the beginning.

2c. Program organizer state the reason, why oral selection again.

4a. Area manager can be cancel loan request after investigation.

Use Case: Saving

Primary Actor: Customer, Program Organizer, Area manager.

Stakeholders and interest:

- Customer: Want confirmation for saving.
- Program Organizer: Want to know the requested member has any group.
- Area manager: Want to know the requested member has any group.

Preconditions: The member must have assigned to a group.

Success Guaranteed (Post condition): Confirmation about saving.

Main Success Scenario (or Basic Flow):

1. Customer request to P.O for saving.
2. Program Organizer check for the Group.

3. If yes then this request goes to the area manager.
4. Then area manager got the request and check that this member have under any group.
5. If yes then request accept and saving start.

Extension s (or Alternative Flows):

Area manager investigate about the member whether the saving permission given or not.

1-1aCustomer will be cancel request for saving.

1-1b. P.O select request for saving.

1-1c. P.O cancel saving request.

2-2a. Group could be request to cancel s/he's saving request.

2-2b. P.O select request for saving.

2-2c. P.O cancel saving request.

3. Area manager field visit can be postponed.

3a. Area manager start s/he's field visit again.

3b. Area manager state the reason, why this field visit again.

4a. If no group then request cancel.

Use Case: Loan procedure

Primary Actor: Customer, Program Organizer, Area manager.

Stakeholders and interest:

- Customer: Want confirmation for loan amount acceptance.
- Program Organizer: Want to know the requested member has any group.
- Area manager: Want to know the requested member has any group and also have enough saving for loan.

Preconditions: The member must have assigned to a group and enough saving for loan.

Success Guaranteed (Post condition): Acceptance about getting loan.

Main Success Scenario (or Basic Flow):

1. Customer request to the P.O for loan.
2. Program Organizer check for the Group.
3. If yes then this request goes to the area manager.
4. Then area manager got the request and check that this member under any group and also have enough saving for loan.
5. If yes then request accept.

Extension s (or Alternative Flows):

1. Customer will be cancel s/he's request for loan.

1a. P.O select the request.

1c. Request cancel from the request list.

2-3a. If no then request cancel.

3. Area manager field visit can be postponed.

3a. Area manager start s/he's field visit again.

3b. Area manager state the reason, why this field visit again.

4. group will be cancel the loan request for that member

4a. Area manager select the request

4b. Area manager cancel the request from the request list.

5a. If no group then request cancel.

6. Account system can be any time crashed.

6a.Accountant restart the system, log in, and request Recovery of prior state.

6c. System reconstructs prior state.

6-1. System detects anomalies preventing recovery.

2. System signals error to the accountant, record the Error and enter a clean state.
3. Accountant has to resubmit the request.

6d. Accountant may able to get the information for area manager.

Use Case: Approval

Primary Actor: Customer, Program Organizer, Area manager.

Stakeholders and interest:

- Customer: Want approval letter for loan.
- Program Organizer: Want to know the requested for loan.
- Area manager: Want to know the requested loan amount is OK or not.

Preconditions: The member must have enough saving for that loan.

Success Guaranteed (Post condition): Approval letter for loan.

Main Success Scenario (or Basic Flow):

1. Customer request to P.O for certain amount of loan.
2. Program Organizer send request to area manager.
3. Then area manager got the request and check that propose amount of loan is OK or not.
4. If yes then loan approved.

Extension s (or Alternative Flows):

1. Many members can apply at same time that's why request can be cancel for temporarily.
2. Program organizer can identify problem, which lead to cancel loan approved.

    2a. S/he goes to the field visit

    2b. Gathered information and send to the area manager.
3. Area manager verify information and balance with propose amount, if not matched loan approval cancel.
4. Customer asks area manager to cancel the approved loan.

    4a. Area manager identify approval paper for removal from approved list.

    4b. Area manager cancel loan approve from the approved list.
5. If system fail accountant give loan and manually keep the track, or cancel the loan.
6. Area manager told for change balance.

    6a.Customer request again with new propose amount.

    6b.P.O send request to the area manager

    6c.Area manager again check saving balance with propose loan amount

    6d.For positive result get loan approved letter

7-4a. If no then loan does not approved.

<u>Use Case: Loan disbursement</u>

Primary Actor: Customer, Area manager, System.

Stakeholders and interest:

- Customer: Want to get loan.

- Area manager: Want to check the approved paper with the accountant.

Preconditions: The member must have loan approved paper.

Success Guaranteed (Post condition): Loan disbursement successfully.

Main Success Scenario (or Basic Flow):

1. Customer comes with approved paper to accountant.
2. Area manager check approved paper with the accountant and gives a report to the accountant.
3. Accountant check report with the customer copy
4. If yes then accountant disbursed loan.
5. Loan amount posted to the system and update database successfully.

Extension s (or Alternative Flows):

4a. If the paper not approved then no loan issued for the customer.

Use Case: Installment

Primary Actor: Customer, Program organizer, System.

Stakeholders and interest:

- Customer: Pay back loan installment.
- Program organizer: Collect loan installment.

- System: System take installment input and save.

Preconditions: The member must have a loan.

Success Guaranteed (Post condition): Installment amount taken and inserted in the system successfully.

Main Success Scenario (or Basic Flow):

1. Customer comes to pay back weekly loan installment.
2. P.O takes installment, keep track manual sheet and gives entry in members pass book.
3. P.O gives the sheet to the accountant for computer posting.
4. Accountant post installment to the computer system and database update successfully.

Extensions (or Alternative Flows):

1. Costumer not comes for pay back loan.

2a. PO forgets to keep track in the manual sheet after taking loan back.

2b. PO did not enter the loan amount in the member's passbook.

2c. PO gives loan installment amount in wrong member.

3. Accountant gives wrong entry.

4. System fails.

<u>Use Case: Saving return</u>

Primary Actor: Customer, Program organizer, System, Area manger, Regional manger.

Stakeholders and interest:

- Customer: Want to get saving return.

- Program organizer: Want to take request.
- System: Update database after giving saving return.
- Area manager: Want to check the account balance with saving and send request to the Regional manager.
- Regional manager: Want to check the report which receives from Accountant and Area manager.

Preconditions: The member must have to saving.

Success Guaranteed (Post condition): Taking Saving return and database update successfully.

Main Success Scenario (or Basic Flow):

1. Customer request to PO to get saving return.
2. PO receive request and send it to the Area Manager and Accountant.
3. Accountant Check balance with saving and send report to combined checking with the Regional Office
4. Regional manager check it and send report to cross check with the Accountant.
5. If yes then give the check the saving balance with loan balance.
6. If saving balance is more than loan balance then return saving.

Extension s (or Alternative Flows):

1. Customer has no saving balance.
2. PO forgets to send the request to Area Manager and Accountant.
5. No saving of the member.
6. If saving is less than loan then not return saving.

Use Case: Death Insurance

Primary Actor: Customer, Program organizer, System, Area manager.

Stakeholders and interest:

- Customer: Want to get Death insurance.
- Program organizer: Want to get the request.
- System: Database update after give death insurance.
- Area manager: check the previous installment record and renew status.

Preconditions: The member must relative to the death member and renew membership properly.

Success Guaranteed (Post condition): Installed and database update successfully.

Main Success Scenario (or Basic Flow):

1. Customer's relative comes to P.O to take Death insurance.
2. P.O sends it to the area manager.
3. Area manager check did they Renew membership properly.
4. After crosschecking give the decision.
5. Accountant post the Death insurance into the database and update successfully.

Extension s (or Alternative Flows):

1.3a. Membership is not renewed.

2.6b. No nominee is selected for member.

<u>Use Case: Delete Member</u>

Primary Actor: Customer, Program organizer, System, Area manager.

Stakeholders and interest:

- Customer: Wants to delete loan account.
- Program organizer: Want to get the request.
- System: Database update after deleting loan.
- Area manager: check the previous installment record and saving status.

Preconditions: The member must pay back the entire loan or have savings equal to unpaid loan.

Success Guaranteed (Post condition): Installed and database update successfully.

Main Success Scenario (or Basic Flow):

1. Customer comes to P.O for deleting loan.
2. P.O sends it to the area manager.
3. Area manager check did he pay the entire loan back or saving is more or equal to the unpaid loan.
4. After checking give the decision.
5. Accountant post the saving and loan information into the database and update successfully. When all amount is zero the delete the loan account of the member.

Extension s (or Alternative Flows):

1.1a. Not pay any installment till now.

2.2a. Lots of overdue loan installment.

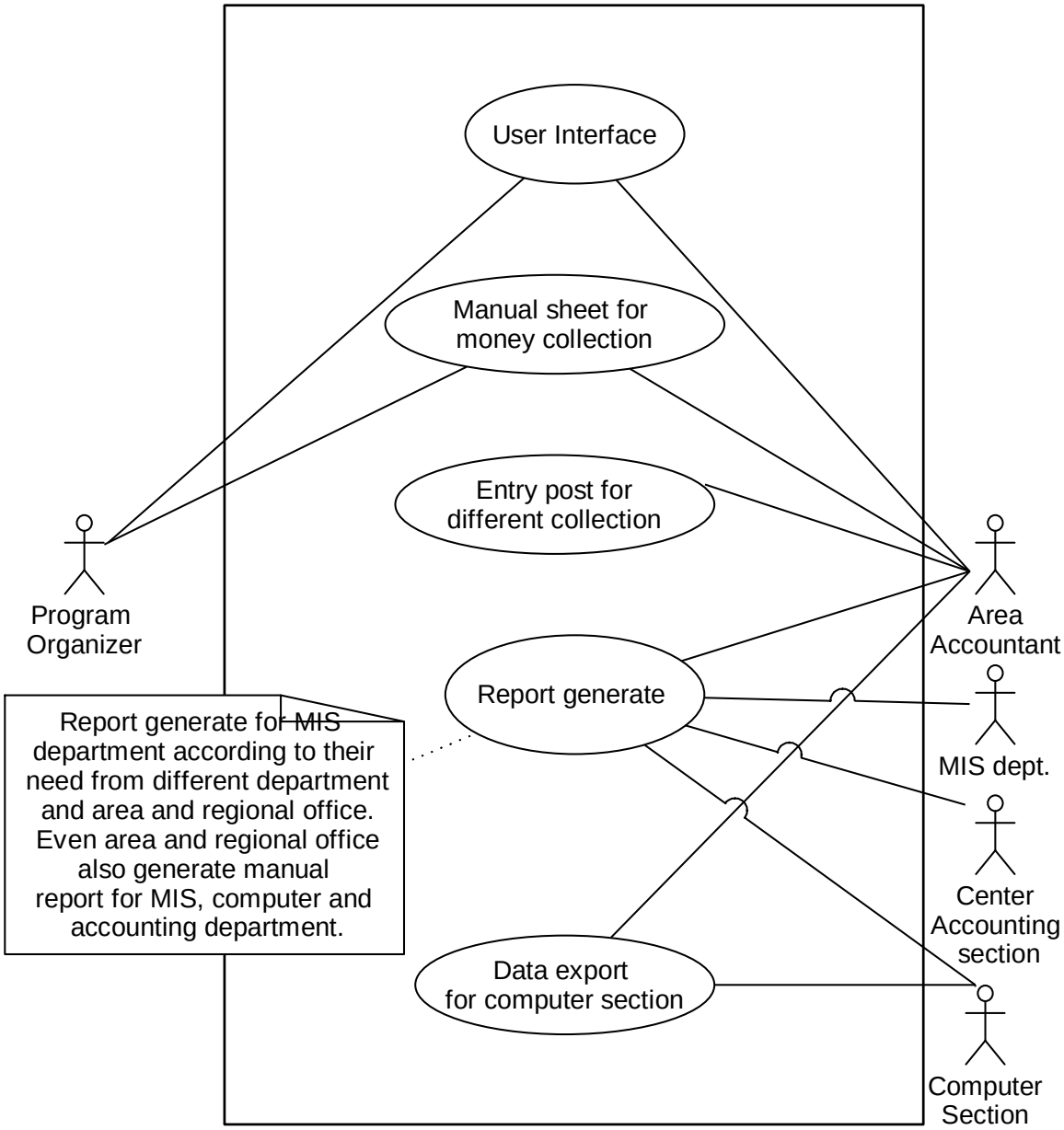


Fig. 4.2 Use case for BRAC micro credit system

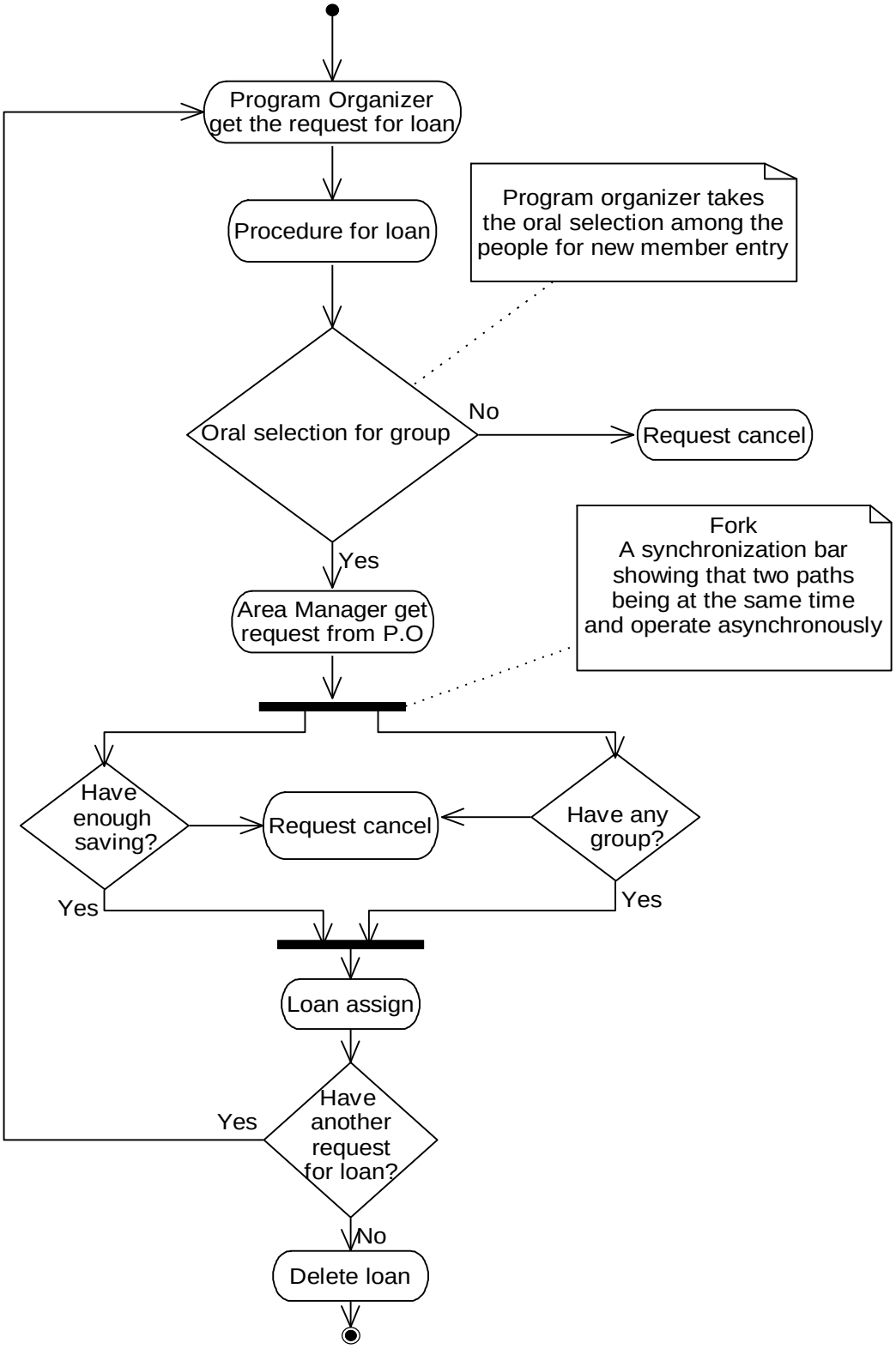


Fig. 4.3 Activity diagram for BRAC micro credit

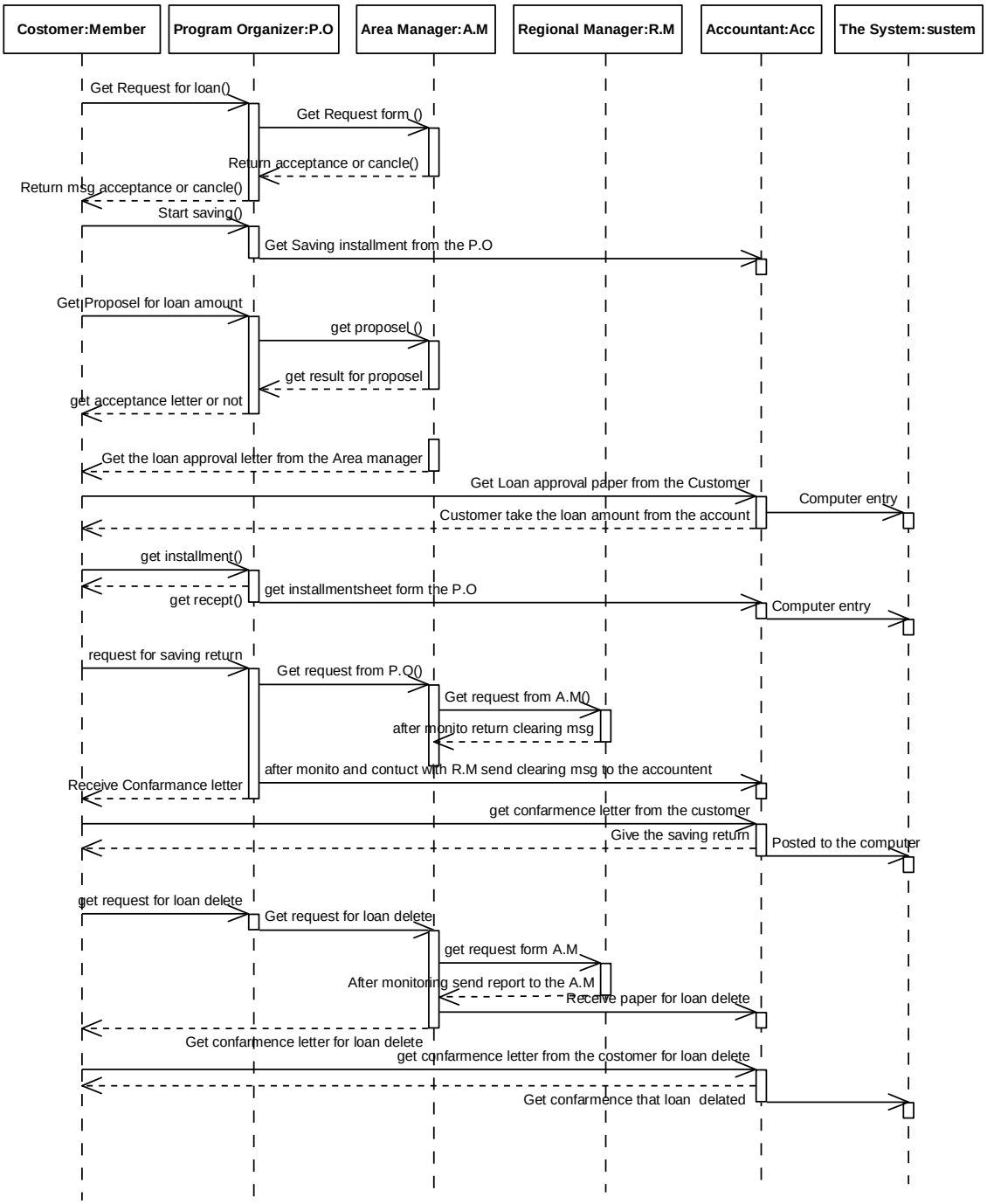


Fig. 4.4 Sequence diagram for BRAC micro credit

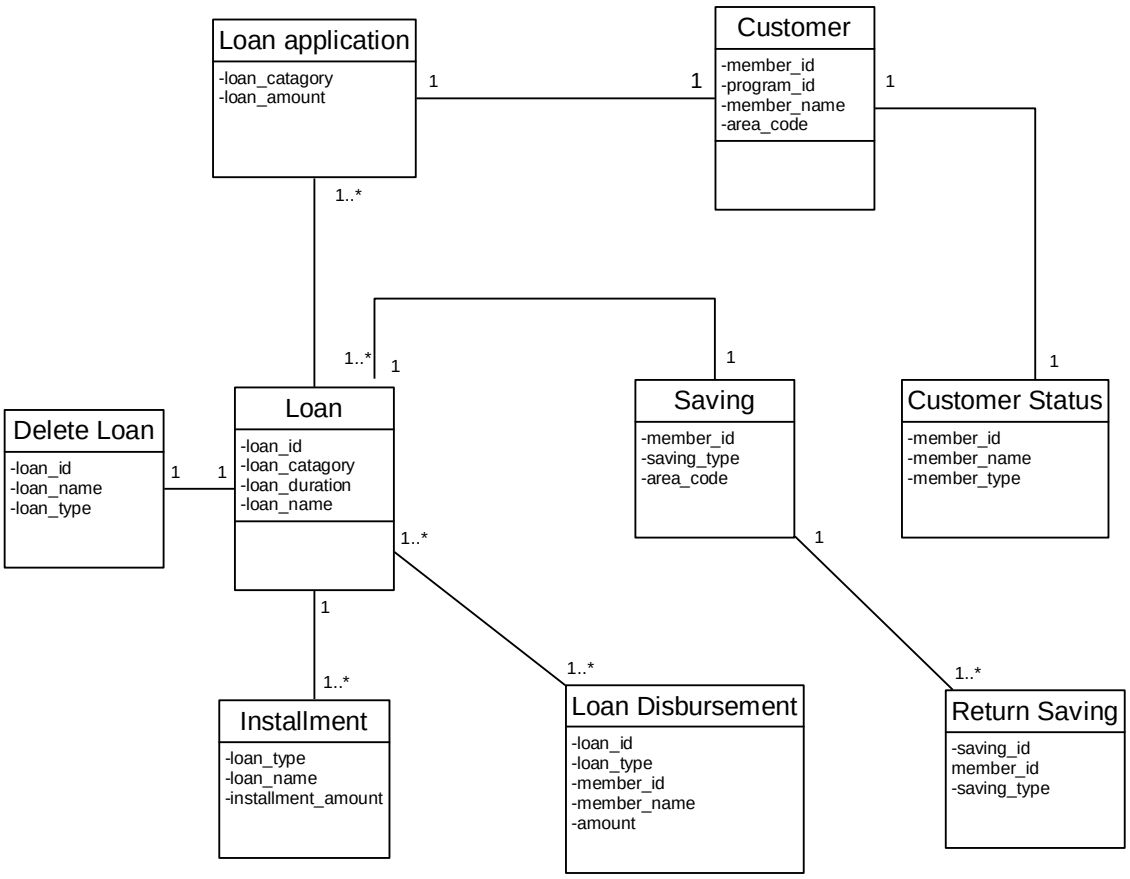


Fig. 4.5 Class diagram for BRAC micro credit

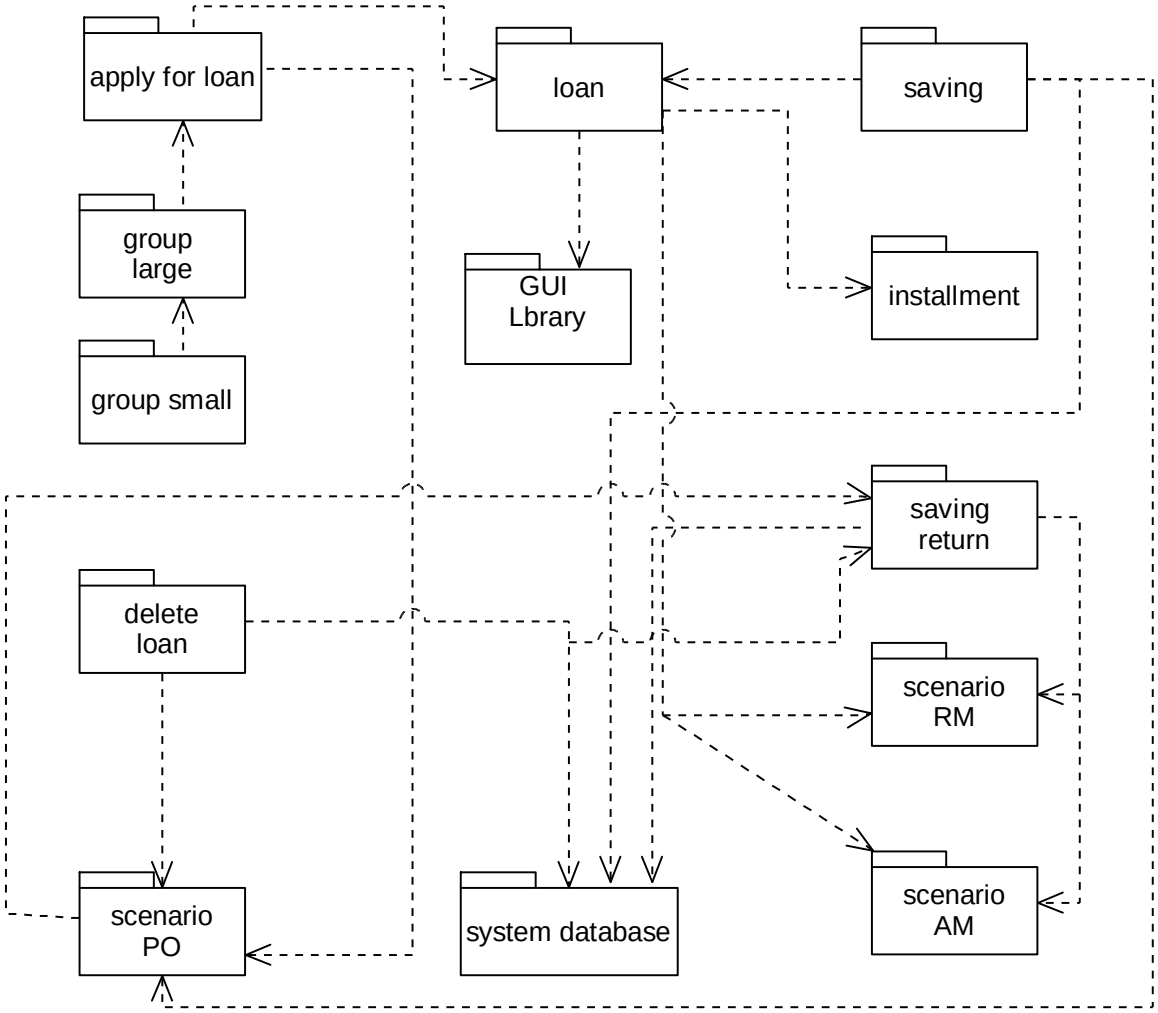


Fig. 4.6 Package diagram for BRAC micro credit

4.1.1 Request for loan:

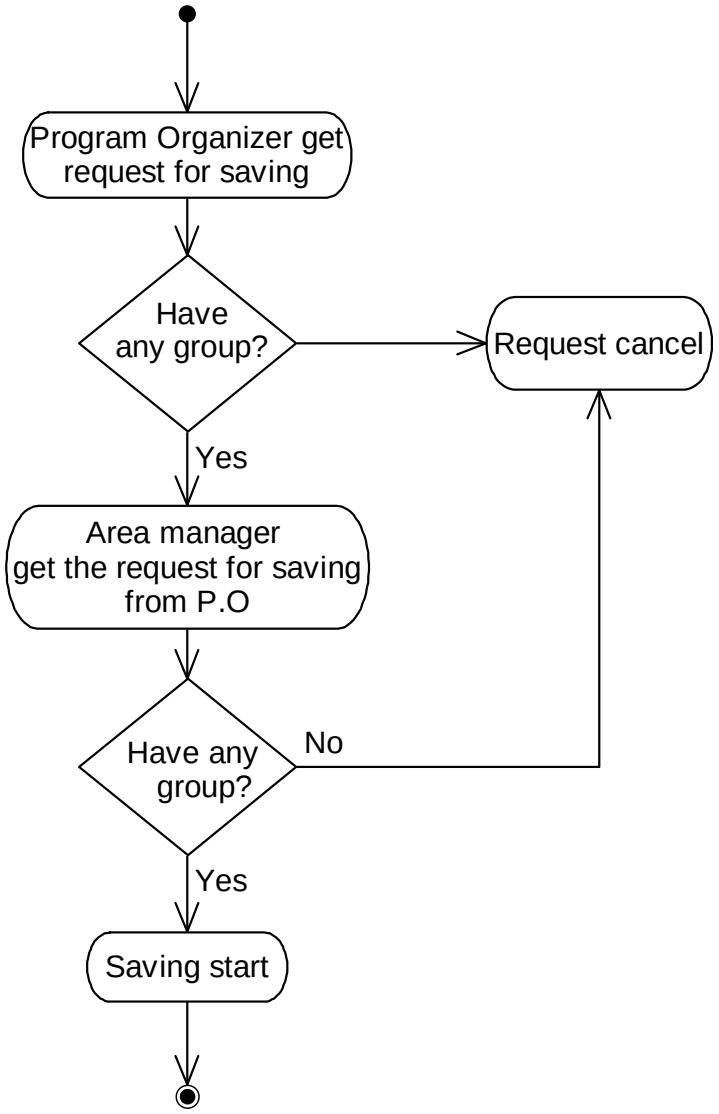


Fig.4.7 Activity diagram for request for Loan (Ref : 2.5.1)

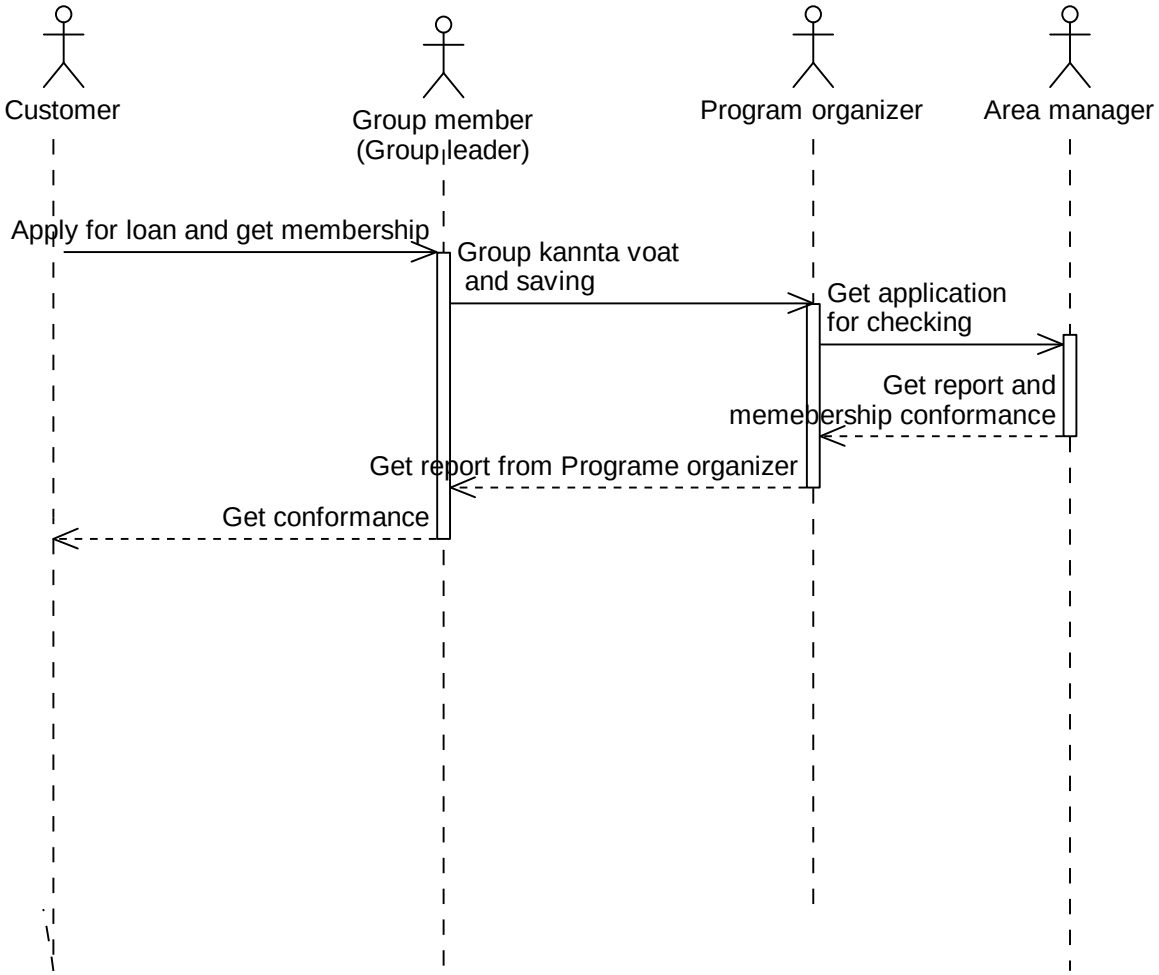


Fig.4.8 Sequence diagram for request for Loan (Ref : 2.5.1)

4.1.2 Savings:

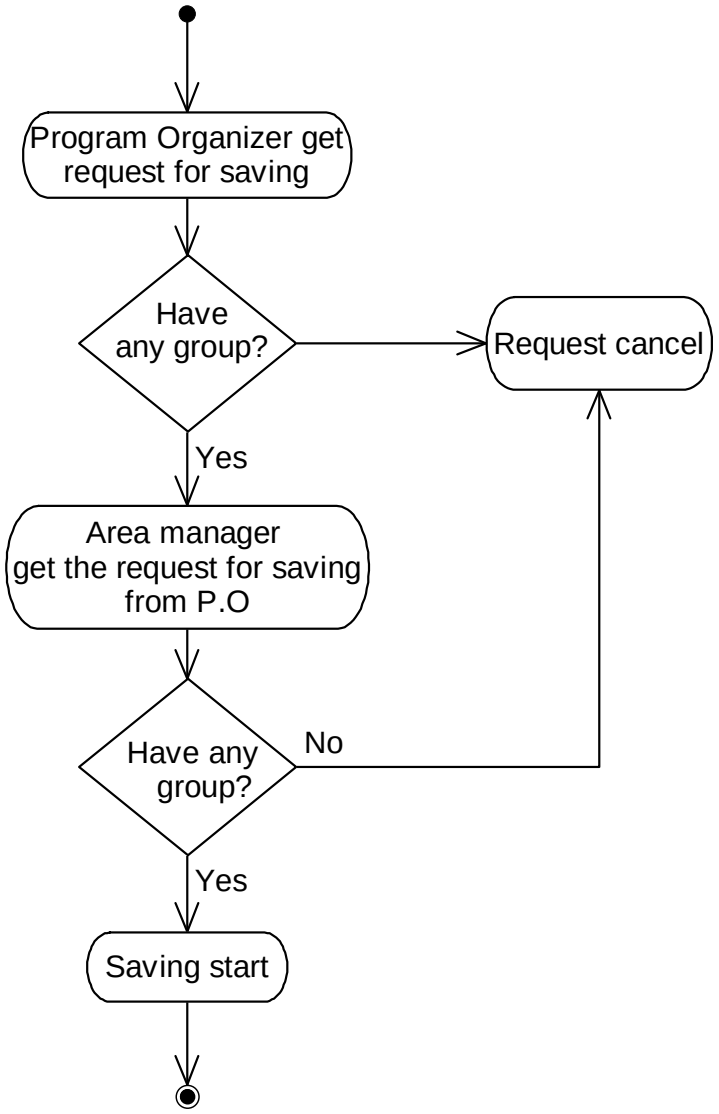


Fig.4.9 Activity diagram for Savings (Ref : 2.5.2)

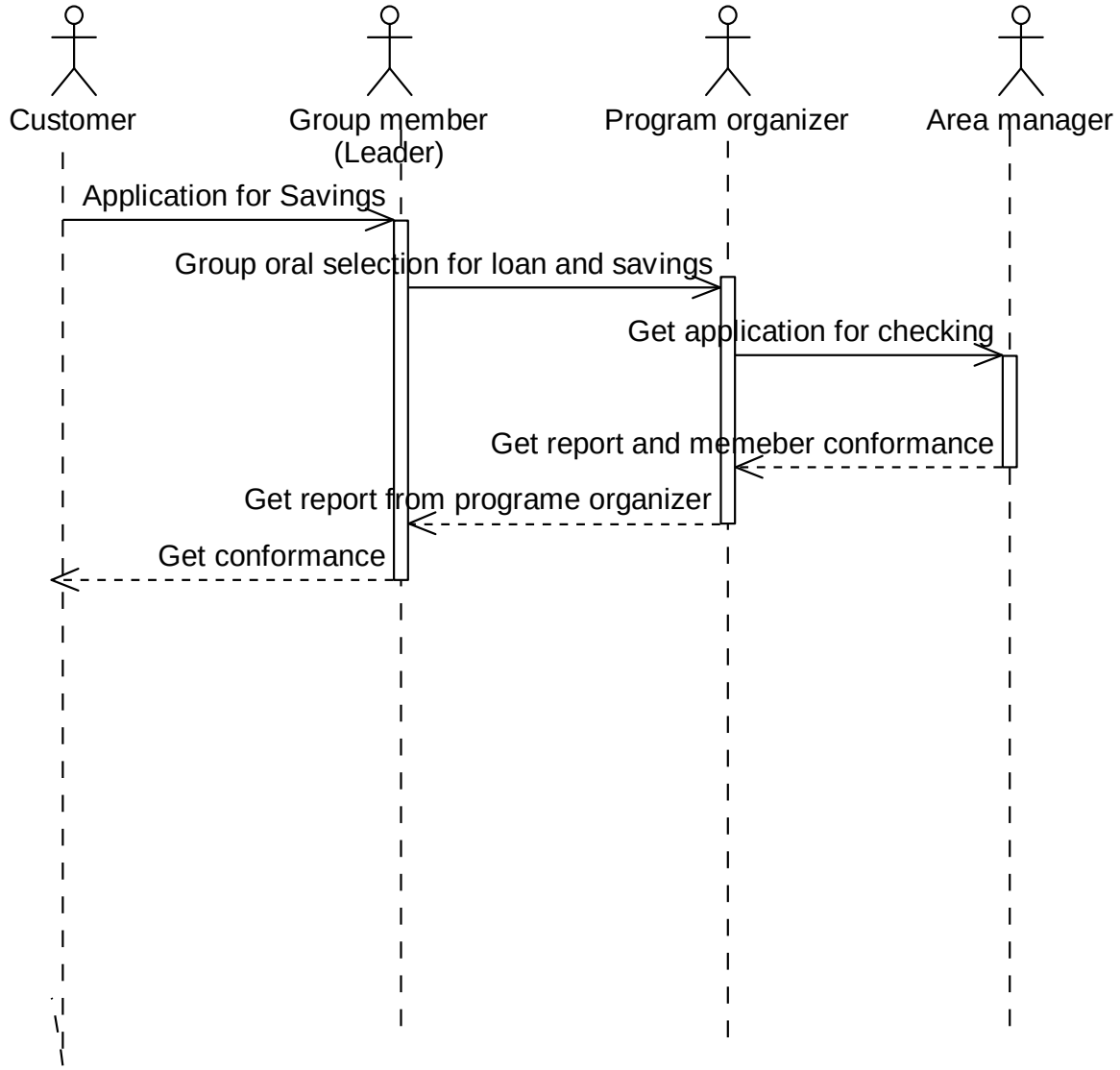


Fig.4.10 Sequence diagram for saving (Ref : 2.5.2)

4.1.3 Procedure for loan:

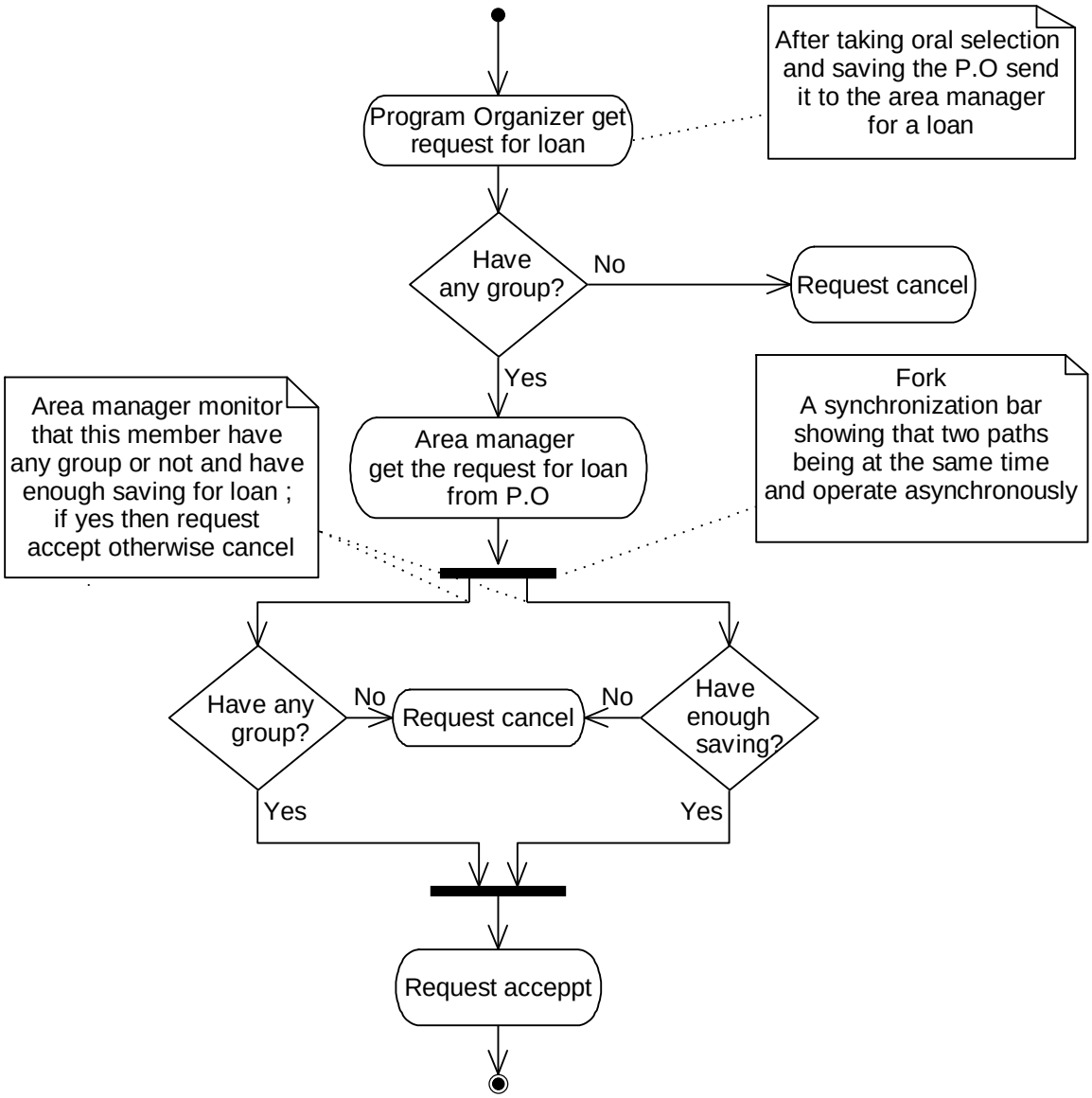


Fig.4.11 Activity diagram for procedure for loan (Ref : 2.5.3)

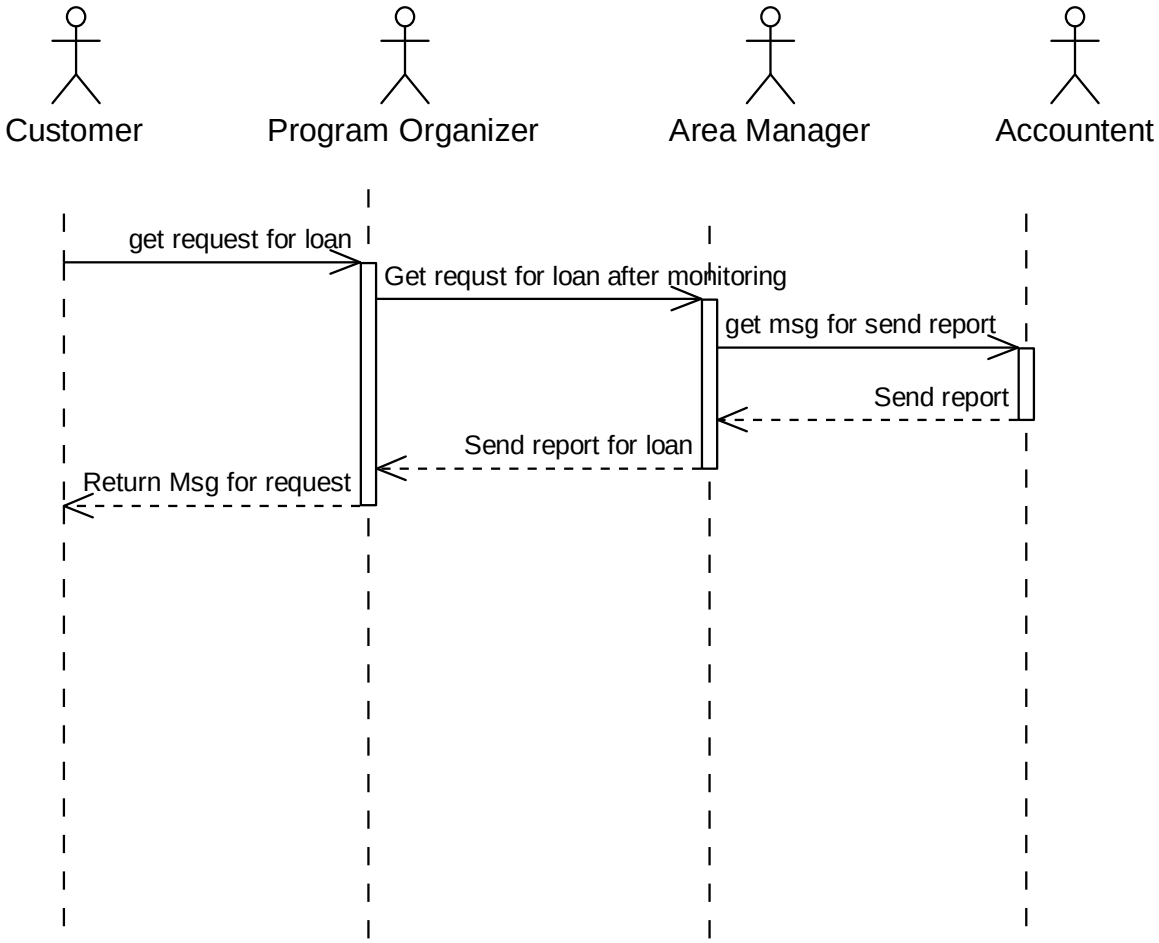


Fig.4.12 Sequence diagram for Procedure for loan (Ref : 2.5.3)

4.1.4 Approval:

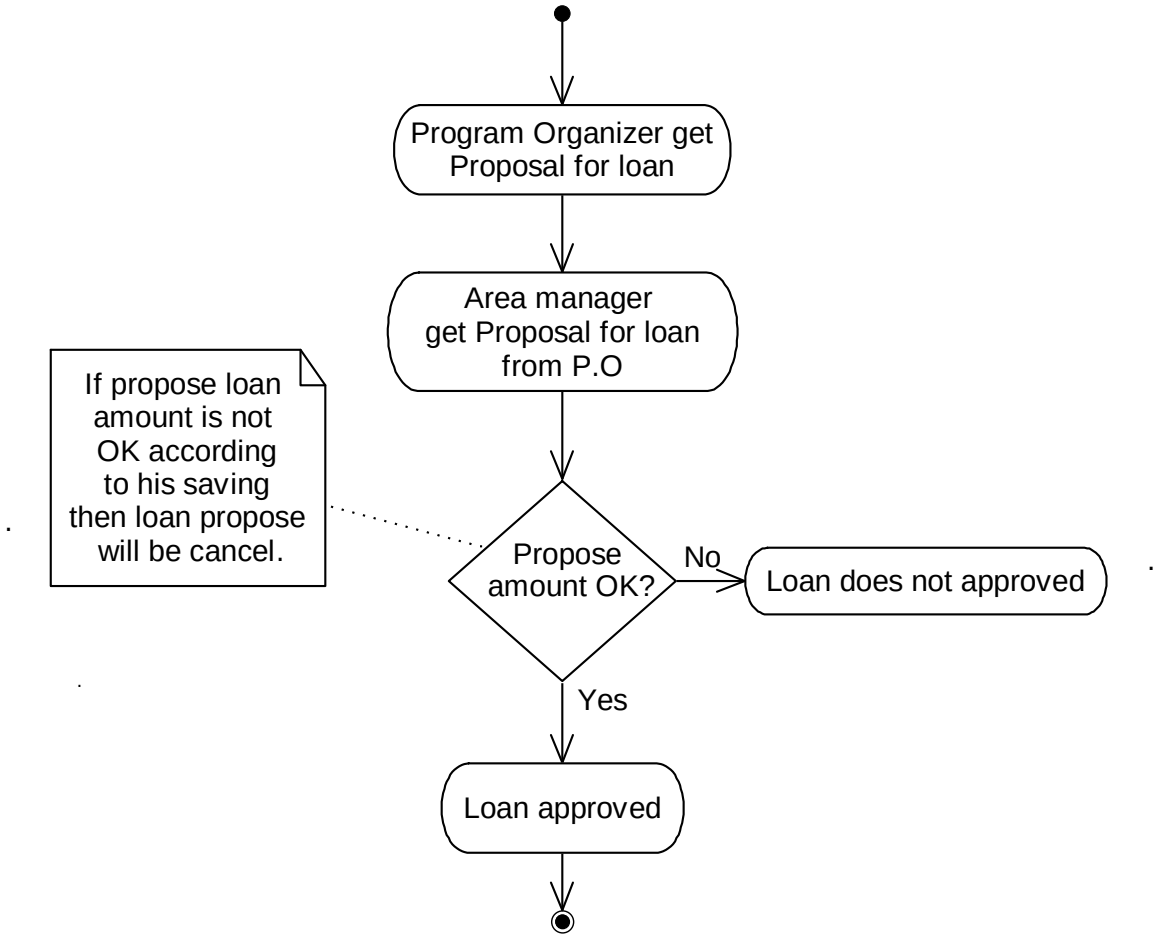


Fig.4.13 Activity diagram for approval (Ref : 2.5.4)

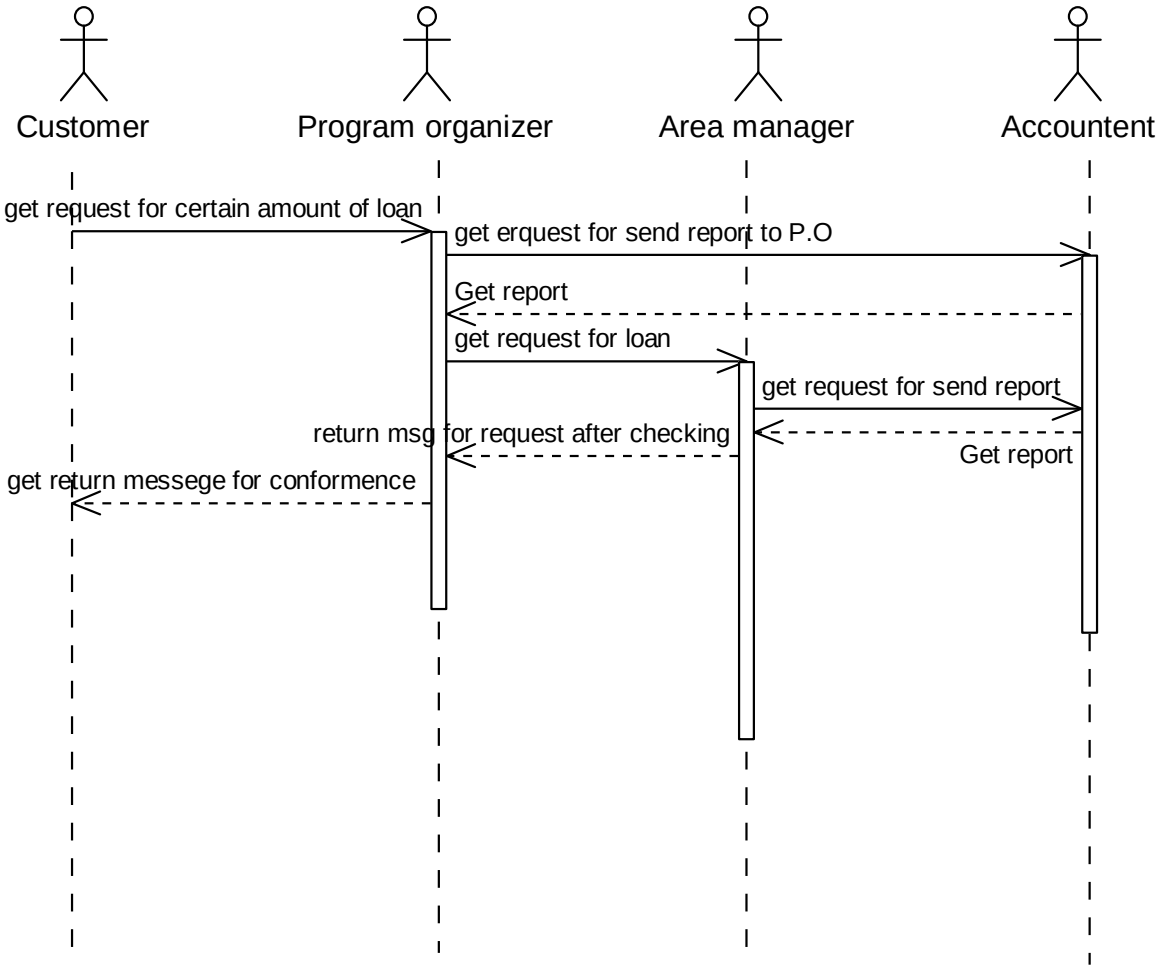


Fig.4.14 Sequence diagram for approval (Ref : 2.5.4)

4.1.5 Disbursement:

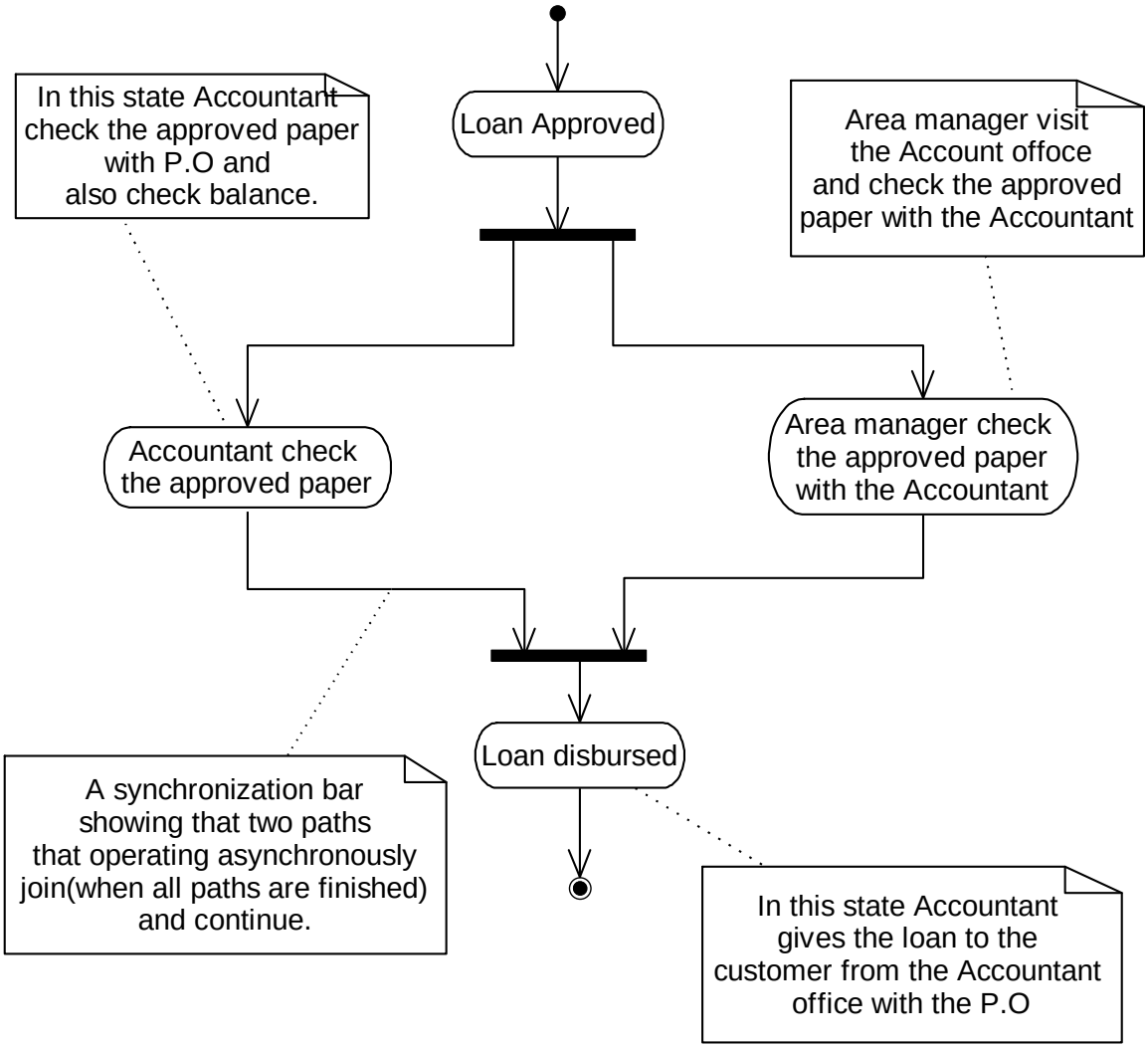


Fig.4.15 Activity diagram for disbursement (Ref : 2.5.5)

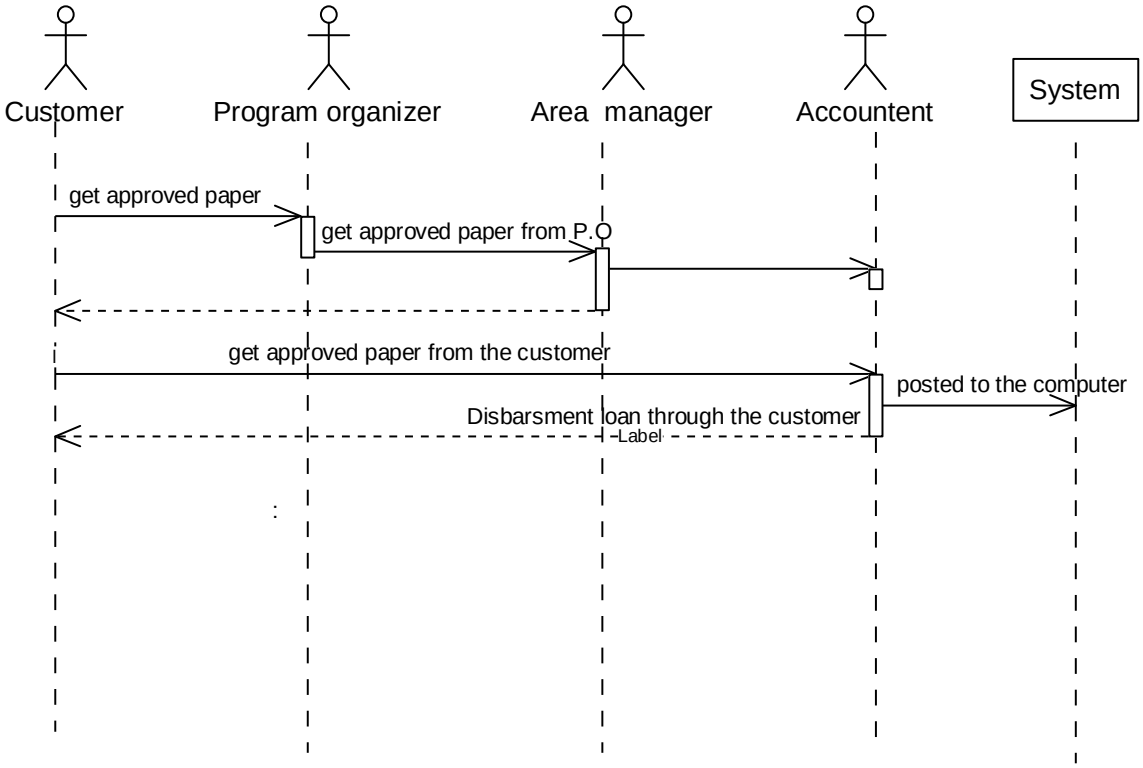


Fig.4.16 Sequence diagram for disbursement (Ref : 2.5.5)

4.1.6 Installment:

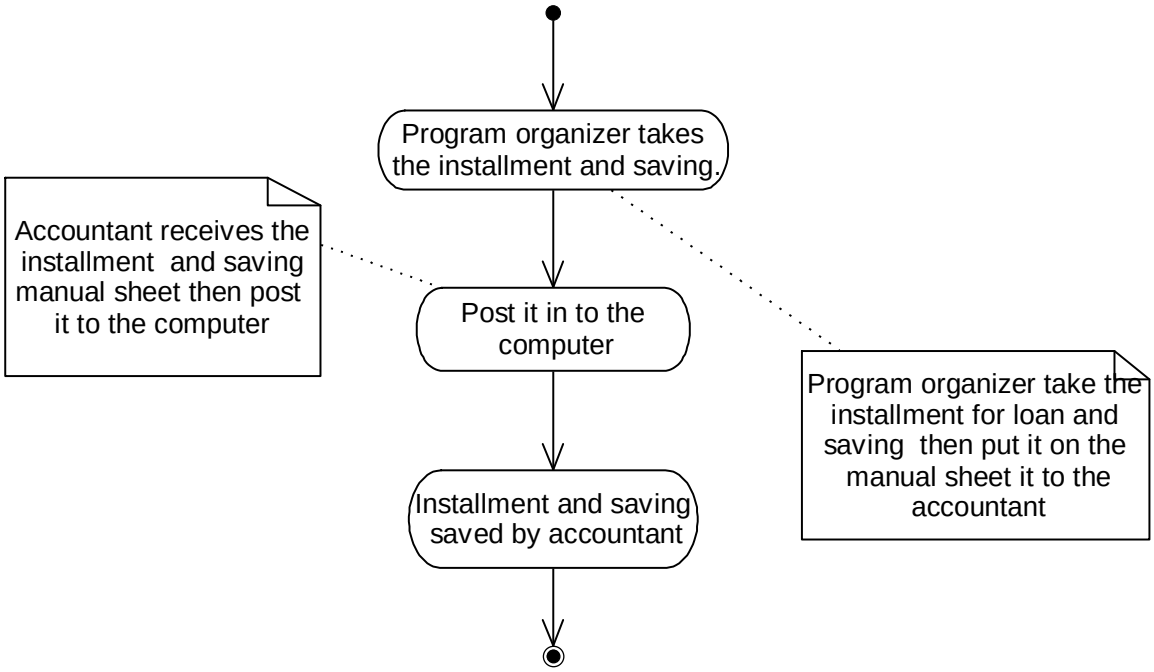


Fig.4.17 Activity diagram for installment (Ref : 2.5.6)

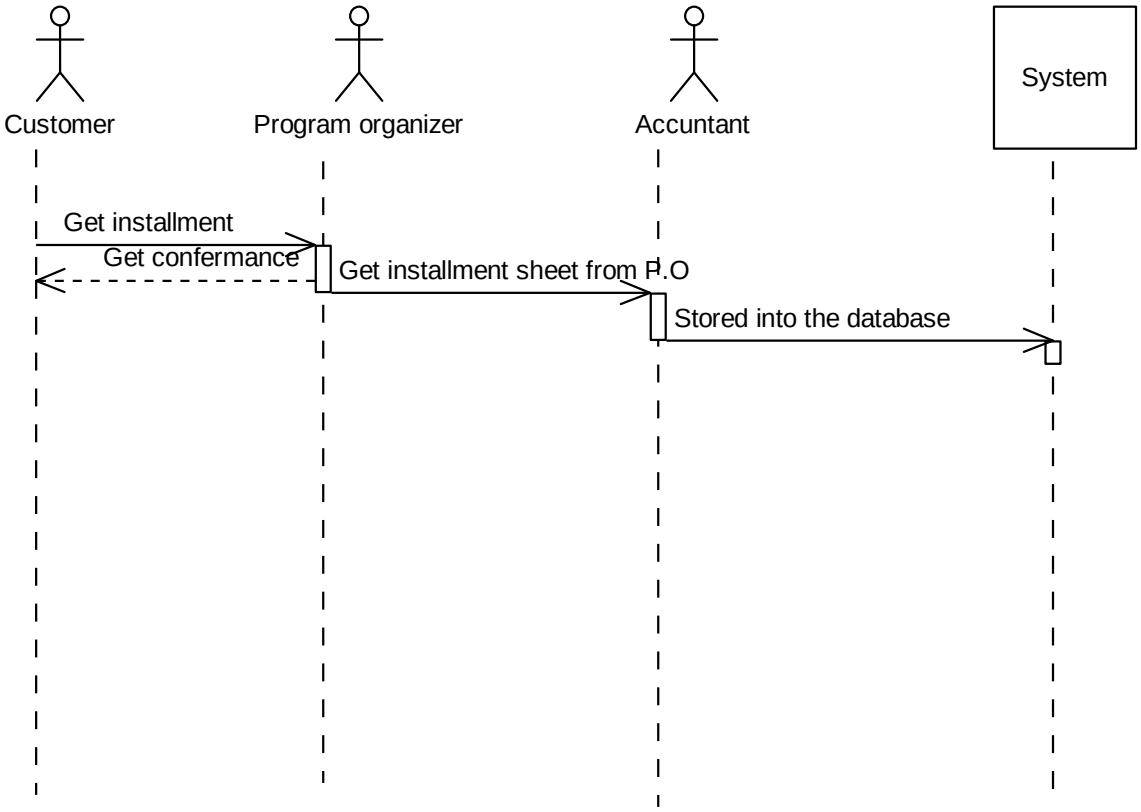


Fig.4.18 Sequence diagram for installment (Ref : 2.5.6)

4.1.7 Saving return:

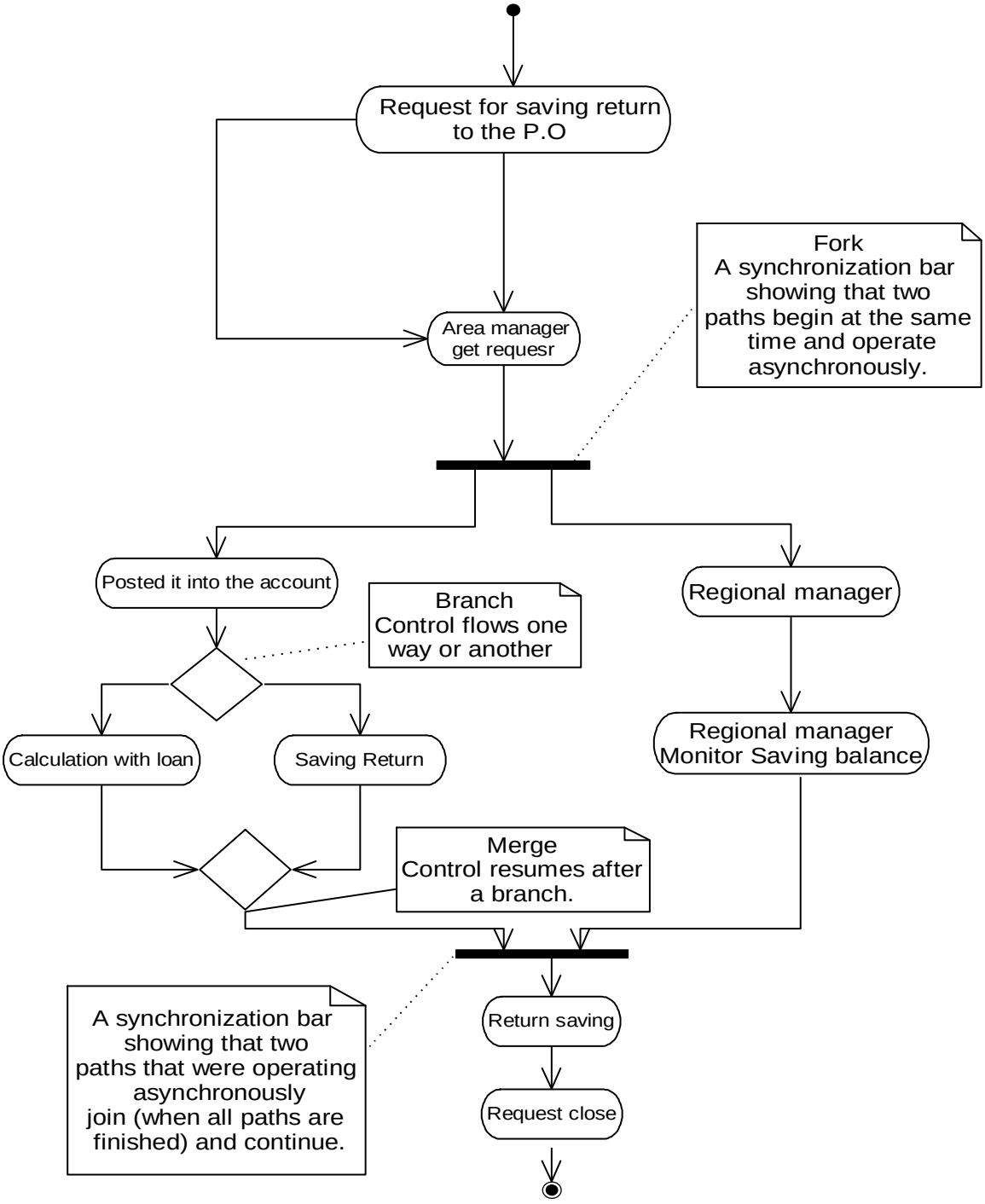


Fig.4.19 Activity diagram for saving return (Ref : 2.5.7)

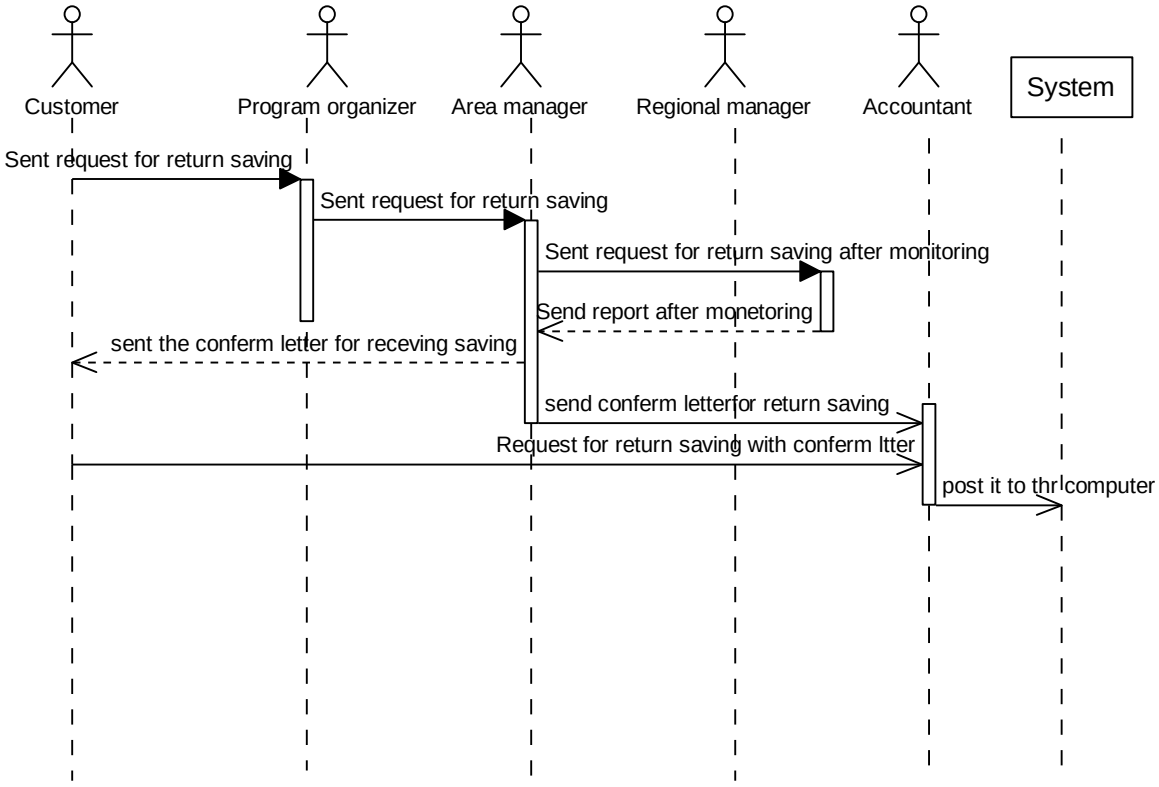


Fig.4.20 Sequence diagram for saving return (Ref : 2.5.7)

4.1.8 Death insurance:

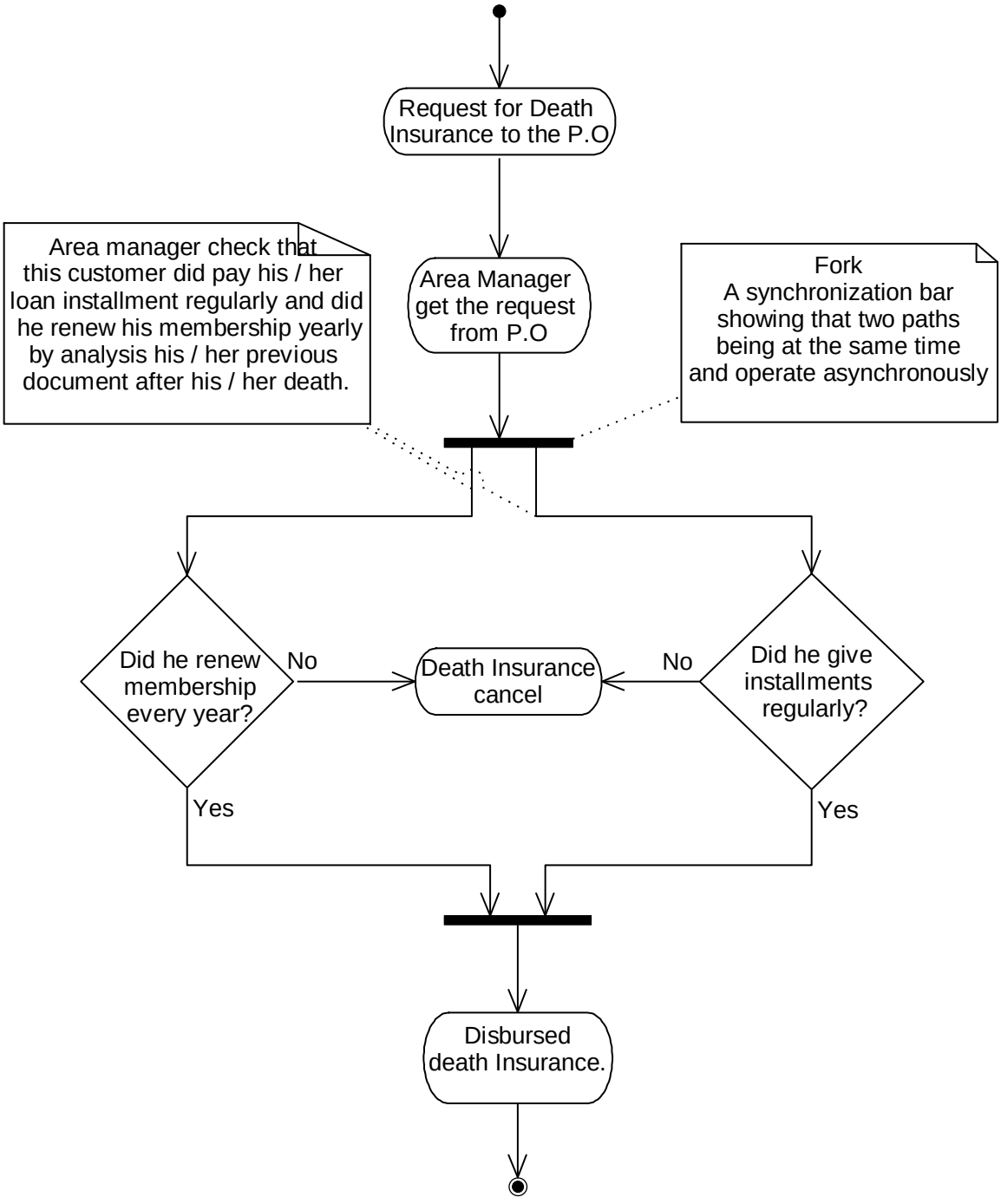


Fig.4.21 Activity diagram for death insurance (Ref : 2.5.8)

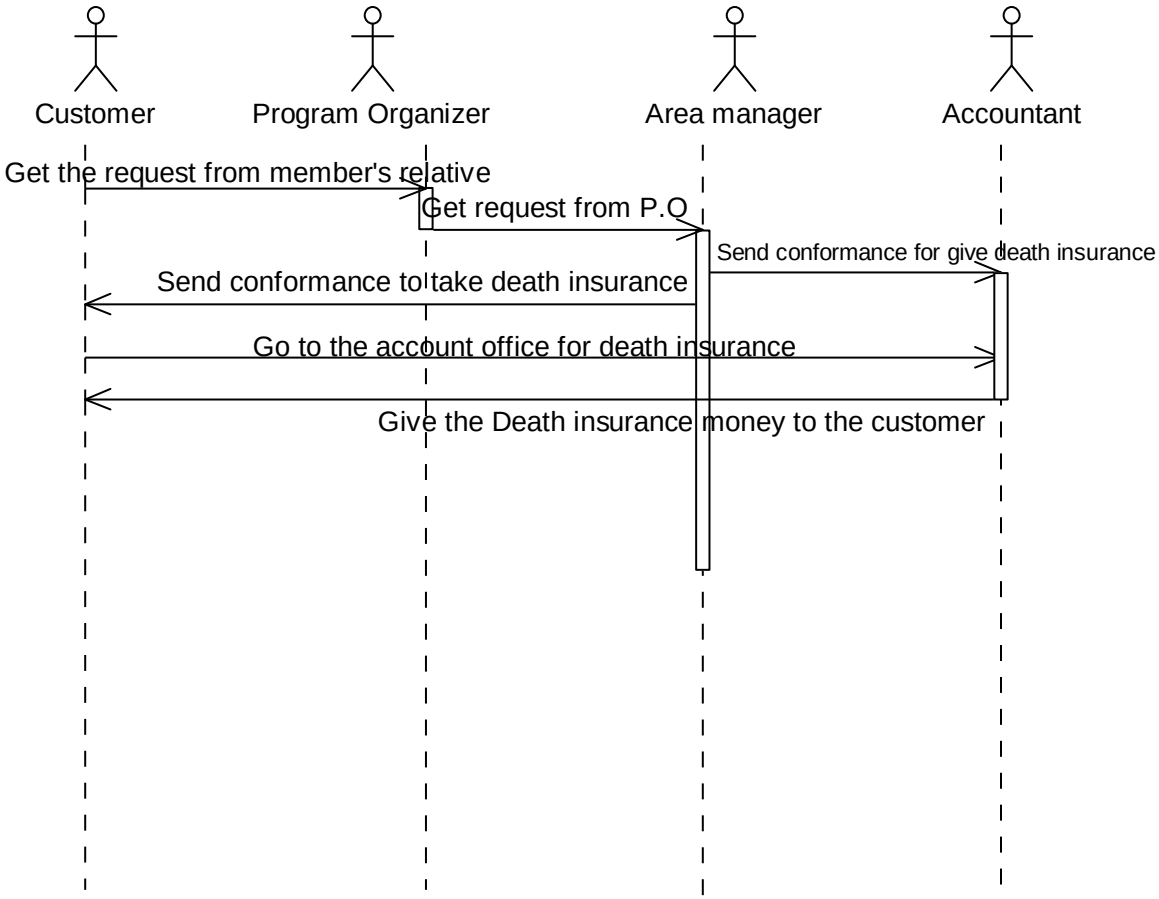


Fig.4.22 Sequence diagram for death insurance (Ref : 2.5.8)

4.1.9 Loan delete:

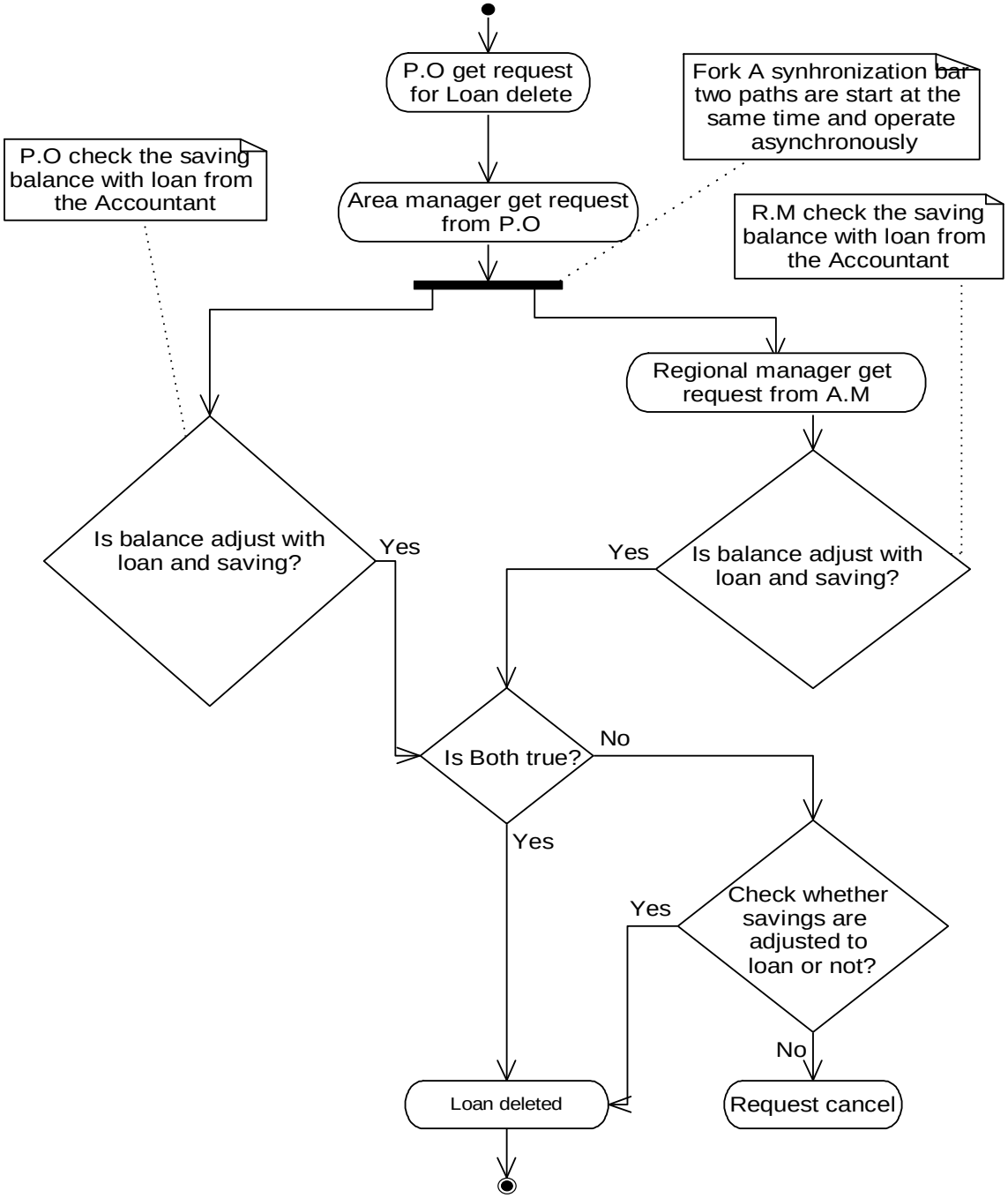


Fig.4.23 Activity diagram for loan delete (Ref : 2.5.9)

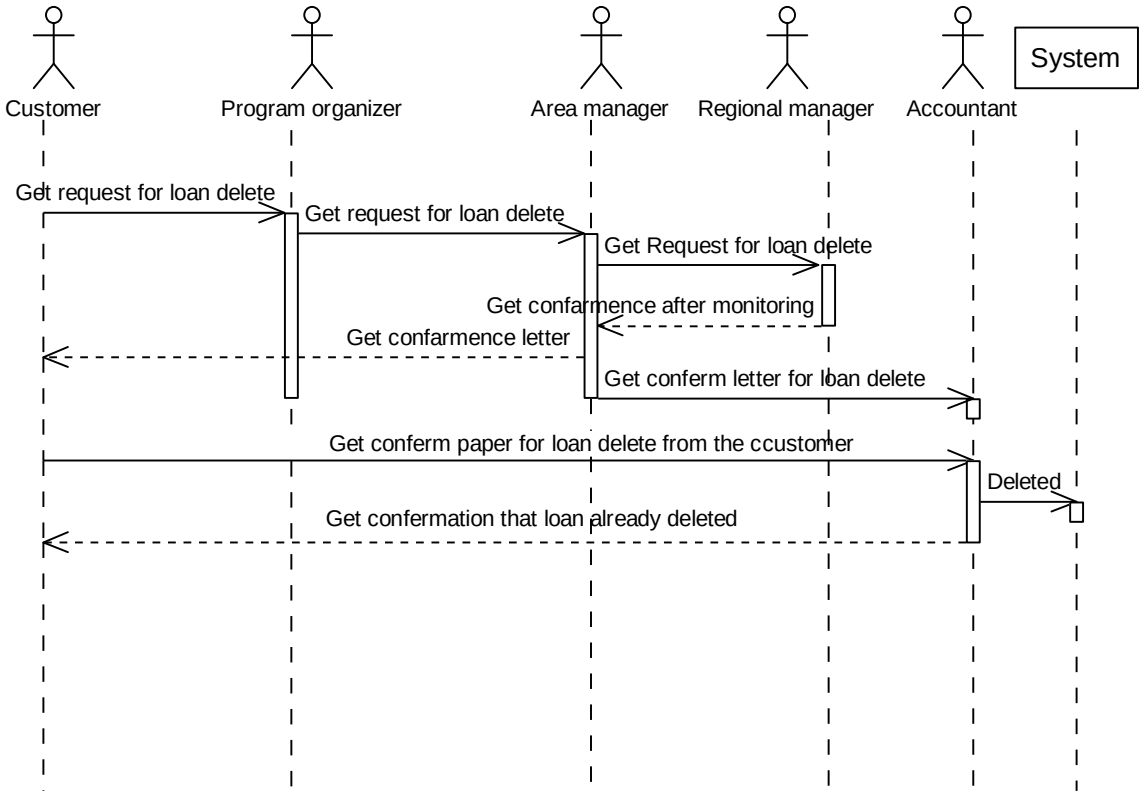


Fig.4.24 Sequence diagram for delete loan (Ref : 2.5.9)

CHAPTER V. PROPOSE SYSTEM

5.1 Proposal 1: Data Flow Trough the Internet from Regional Office to BRAC Center Computer Section

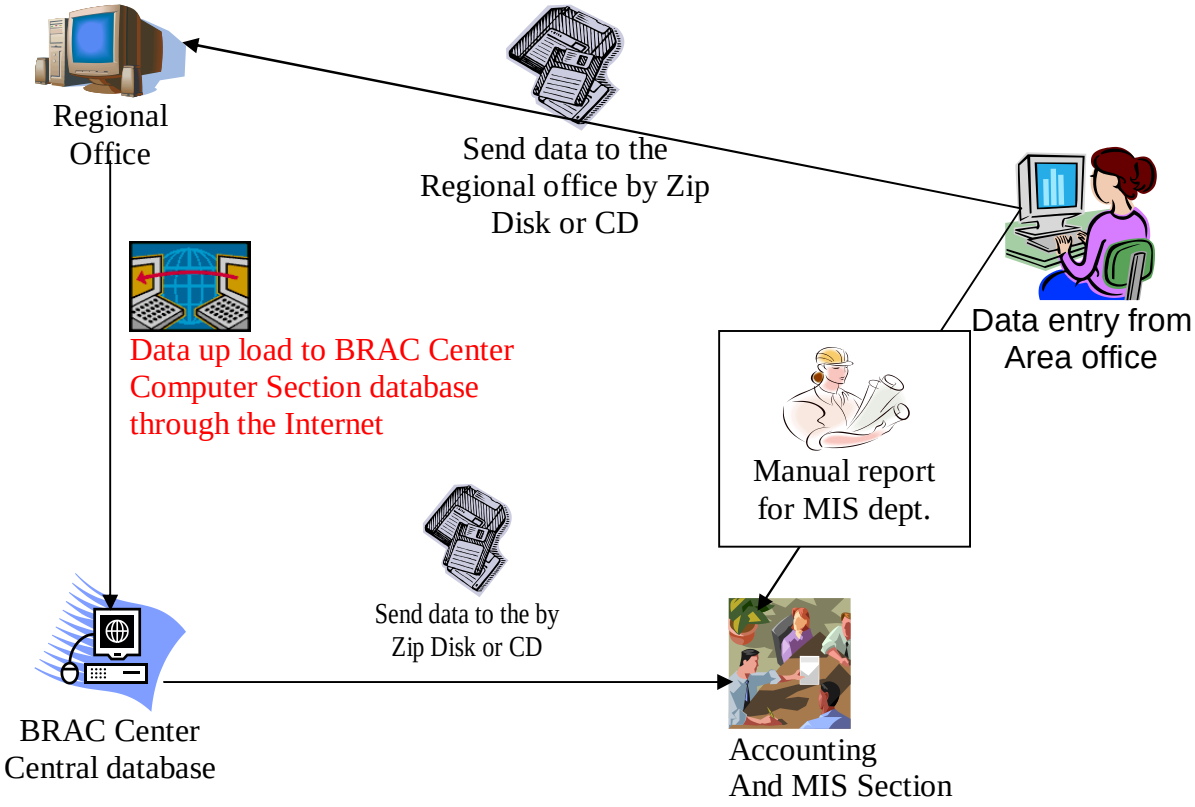


Fig.5.1 Data flow Area office to BRAC Computer section through the Internet.

All data from Area office comes to the regional office by CD or Zip disk. But from the regional office sends all data to the BRAC center computer section through the internet. That's why at first established internet connection from regional office to BRAC center computer section. And also they can do their other information exchange through the internet. Then BRAC center computer section send data to the BRAC center accounting section and MIS department by CD or Zip disk.

By this data flow process save time, money and also increases efficiency.

5.2 Proposal 2: Data Flow From BRAC Center Computer Section to Central Database

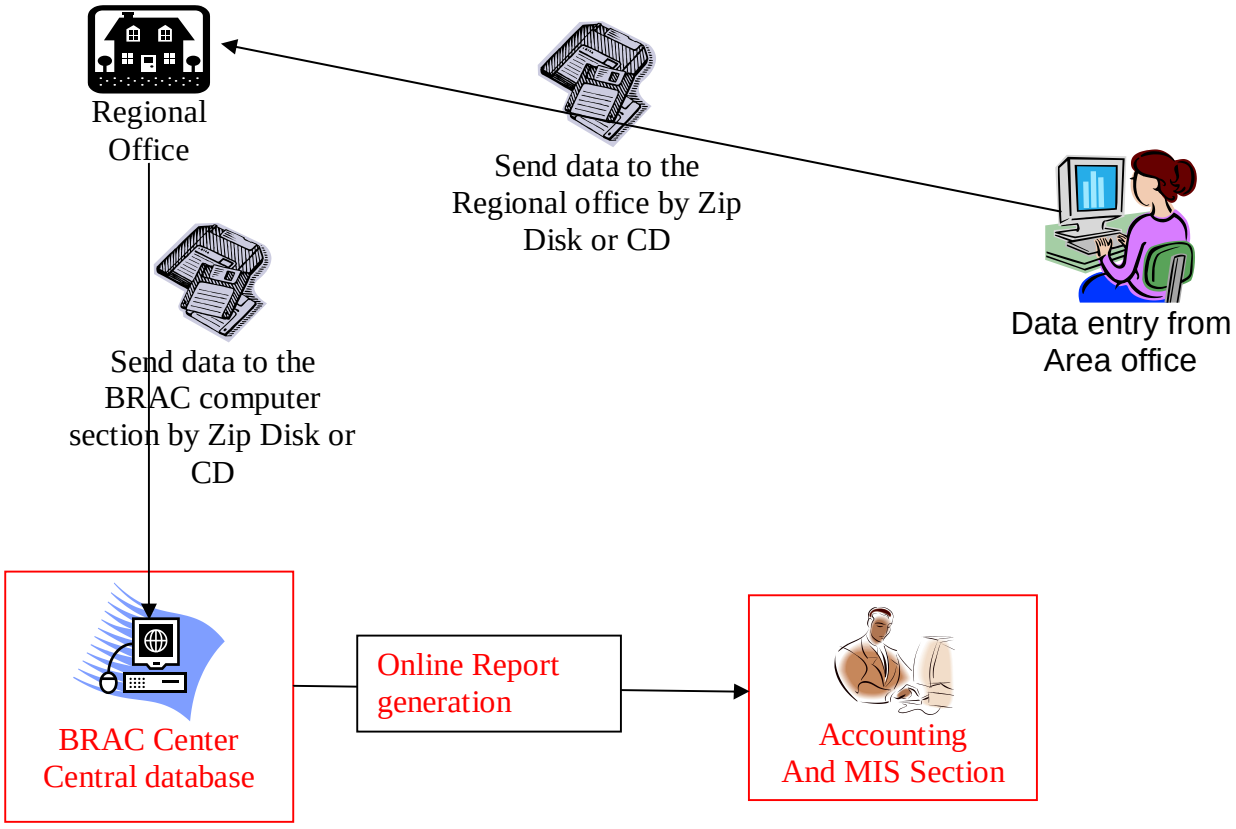


Fig.5.2 Data flow from area office to BRAC Center Computer central database.

This proposal based on the concept of central database where MIS, BRAC computer section, and Accounting section are connected.

BRAC center computer section received all data by their traditional way but from the BRAC center computer section operator upload data to the central database, which can also access BRAC center MIS department, accounting section. For this reason both department can crate any kinds of report any time without any request to the BRAC center computer section. Which lead Flexibility, time saving and also save money.

5.3 Proposal 3. Flow For Correction of "Wrong entry"

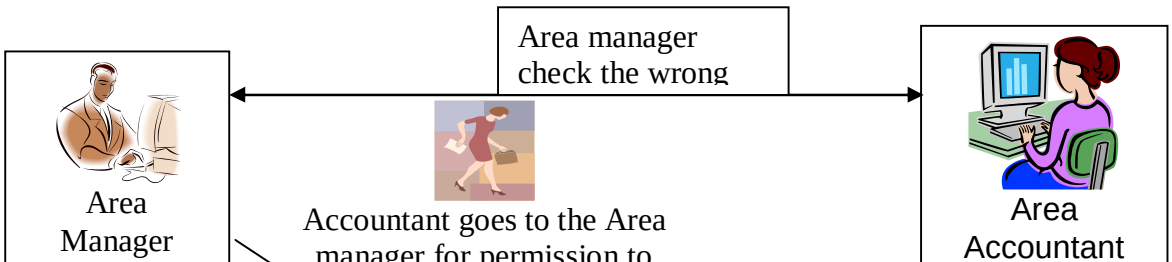


Fig.5.3 Propose steps for data error correction

For wrong correction it is difficult time expensive and also inefficient. In this proposal Area accountant don't need to go BRAC center computer section to correct wrong data entry. Area accountant goes Area manager for permission to correct wrong data entry. Then area manager goes to the accountant office to investigation this request. If request is valid then Area manager give permission. Accountant goes to the Regional manager with area manager permission paper. Then Regional accountant edit those wrong data entry. For this edit all responsibility takes Area manager.

That's why area accountant don't need to go BRAC center computer section. Which lead to save time, money, and also increases effectiveness, and efficiency.

5.4 Redefining Processes

5.4.1 Use case:

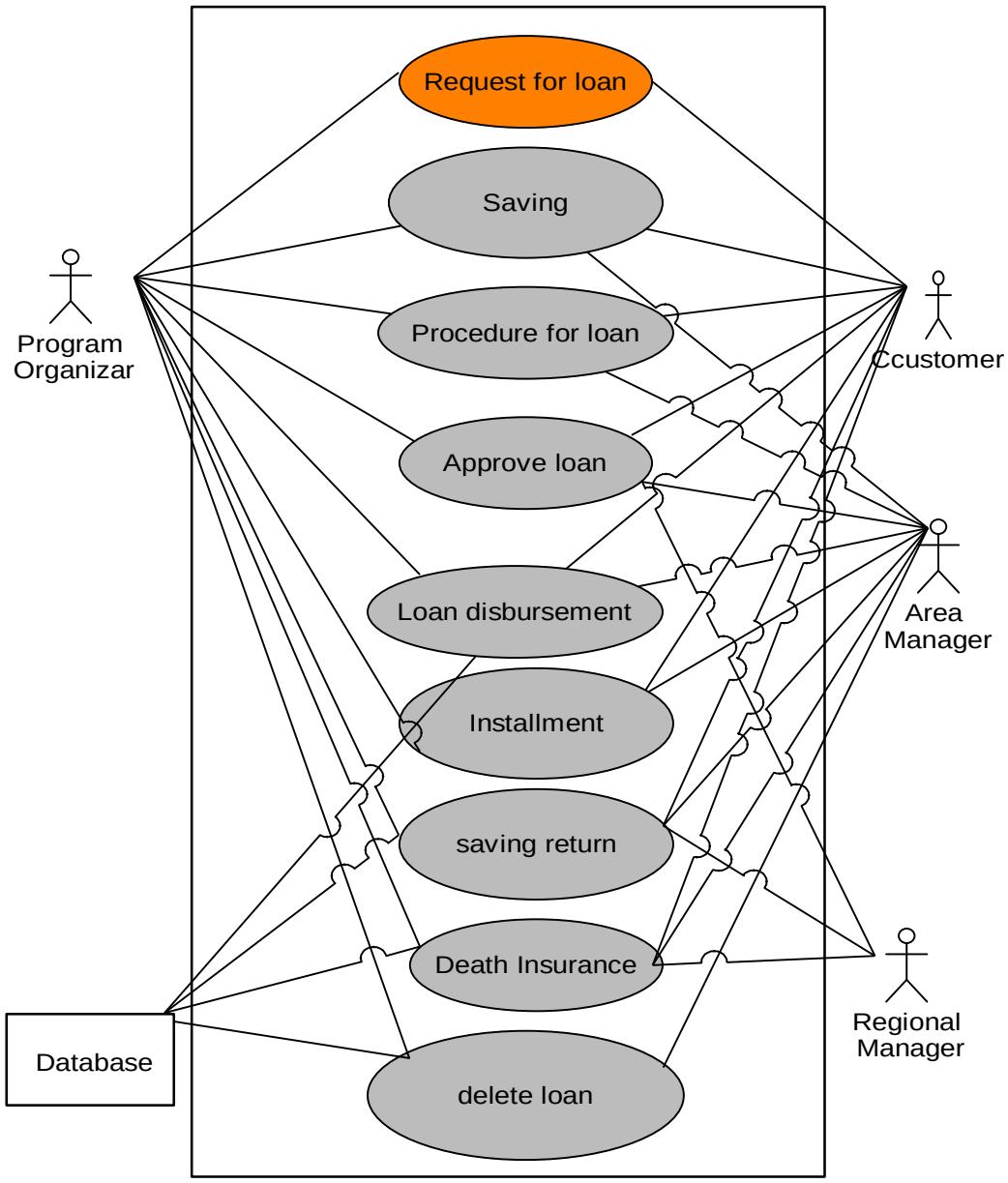


Fig. 5.4.1.1 Redefined use case for BRAC micro credit (Ref:2.2)

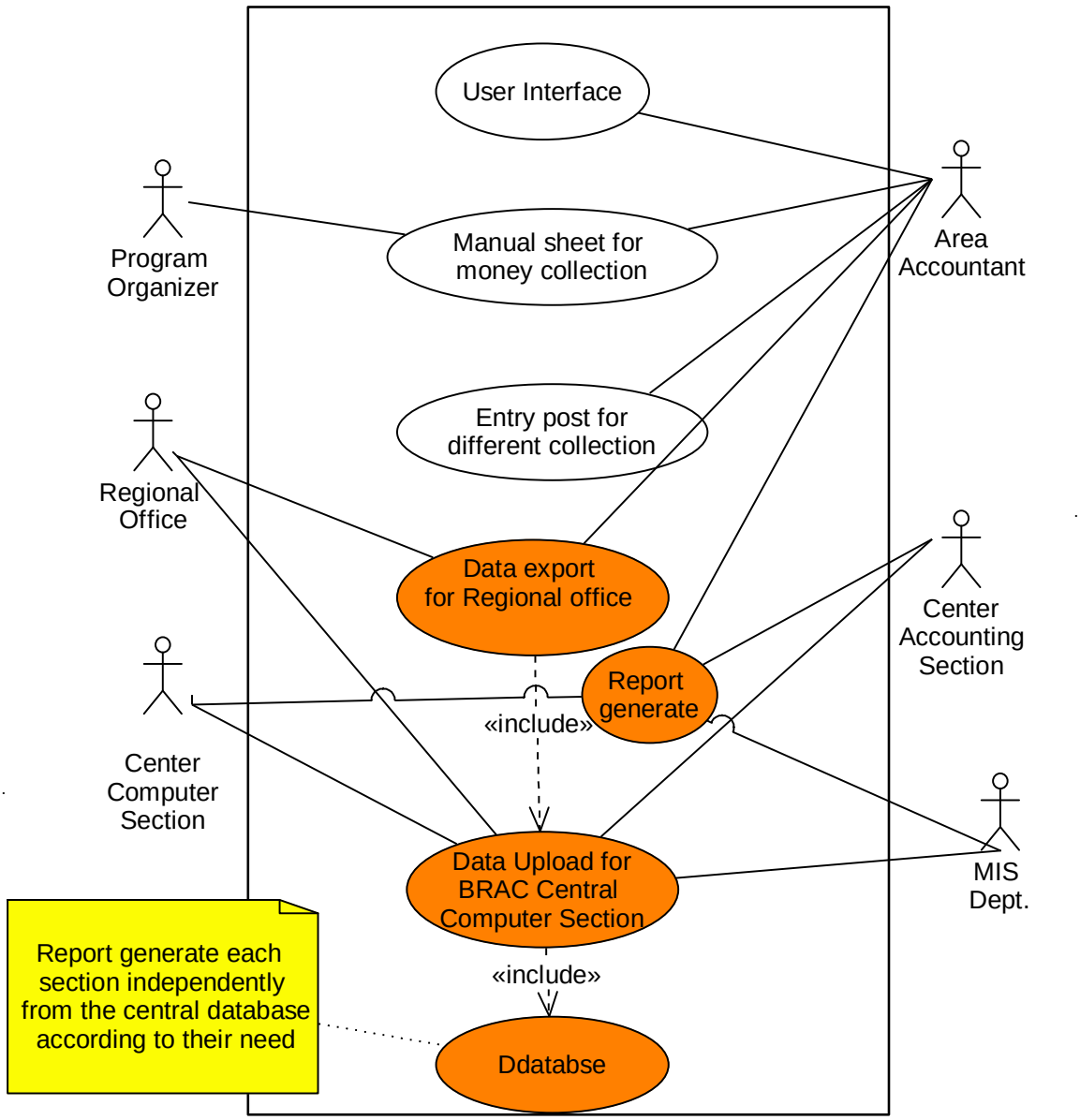


Fig. 5.4.1.2 Redefined use case for BRAC micro credit through the internet

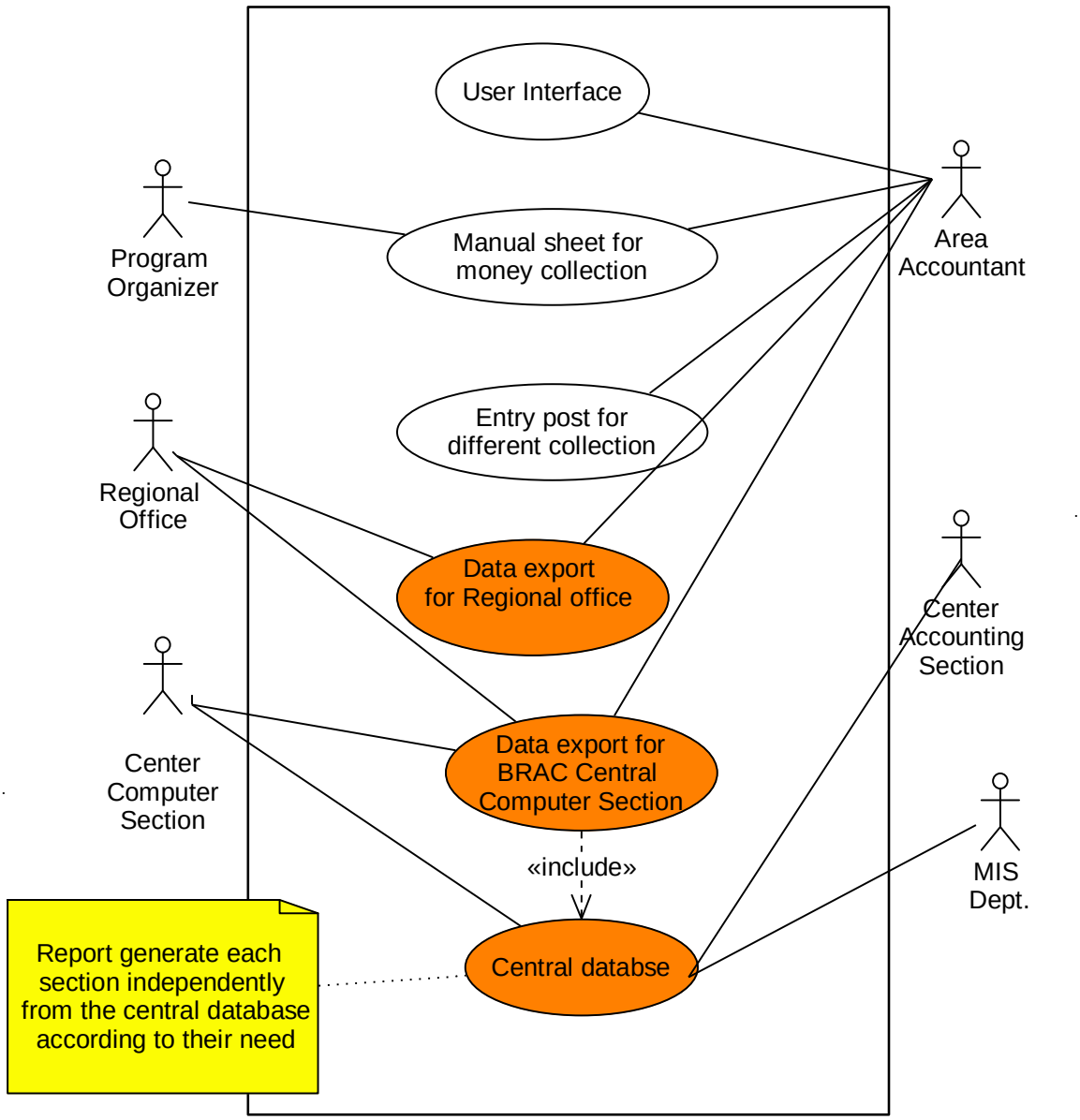


Fig. 5.4.1.3Redefined use case for BRAC micro credit system based on central database (Res: 2.3)

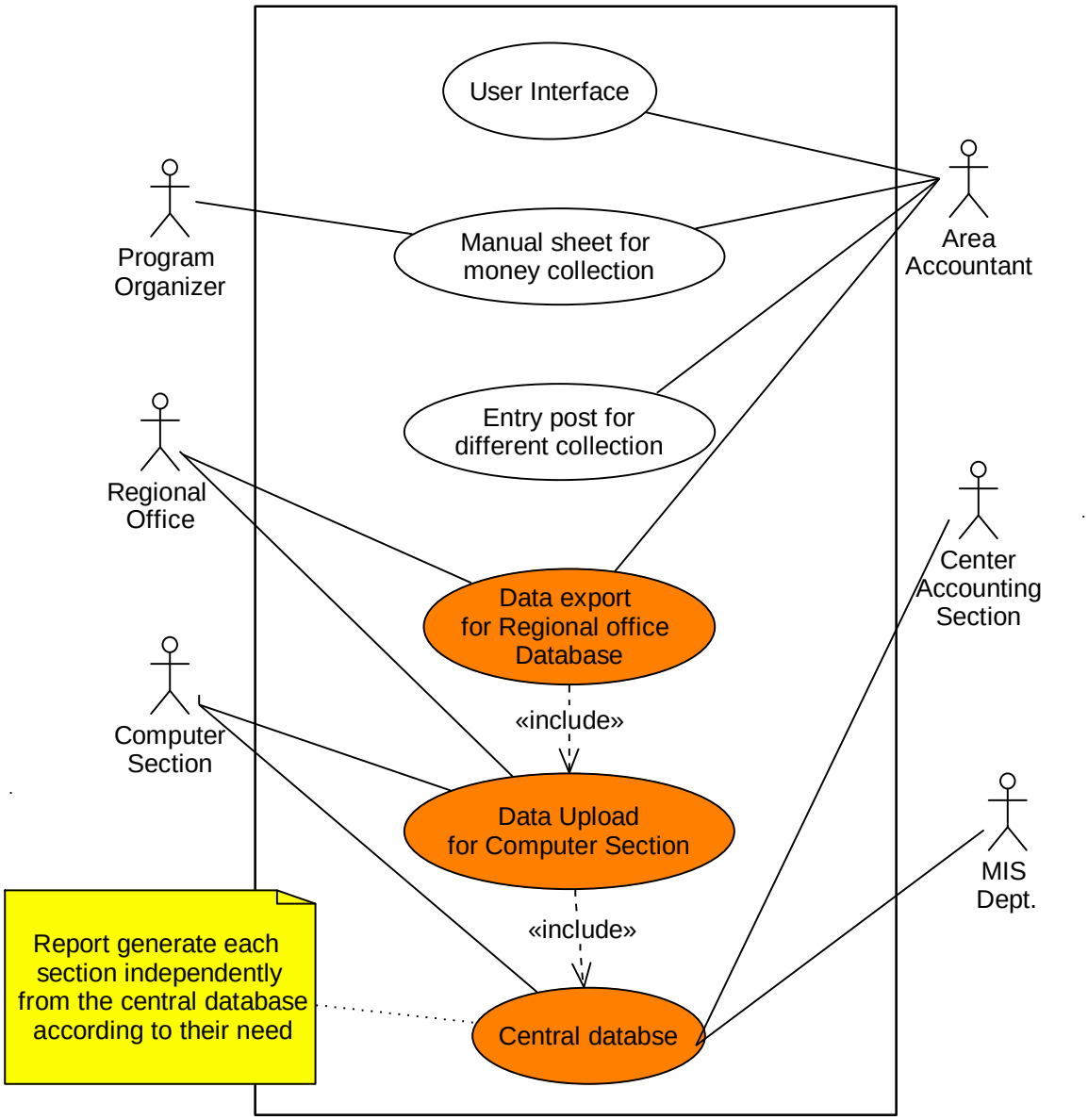


Fig. 5.4.1.4 Redefined use case for BRAC micro credit system Internet and Central database

5.4.2 Activity diagram:

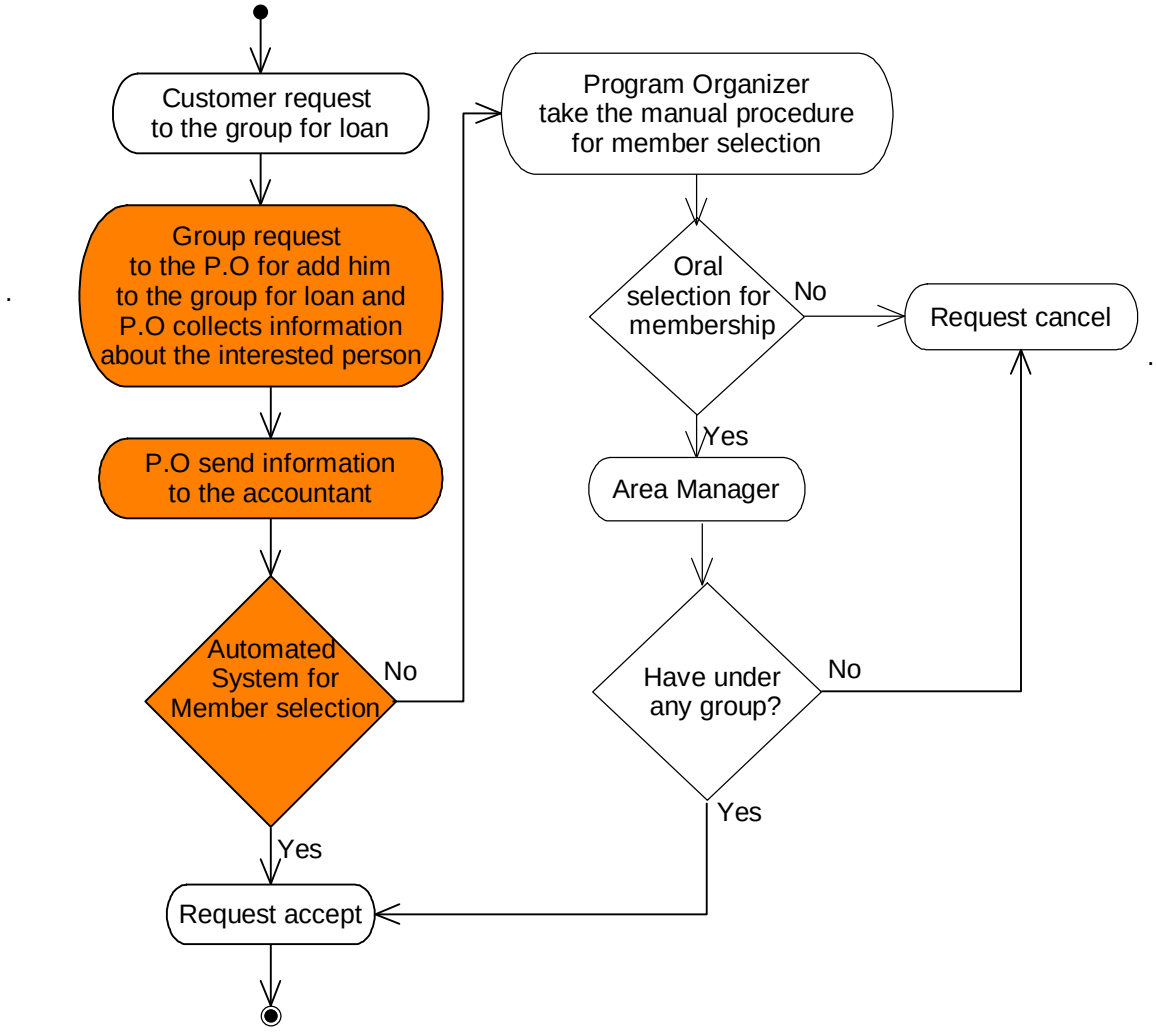


Fig.5.4.2.1 Redefined activity diagram for request for loan (Ref. Fig. 3.0)

5.5.3 Sequence diagram:

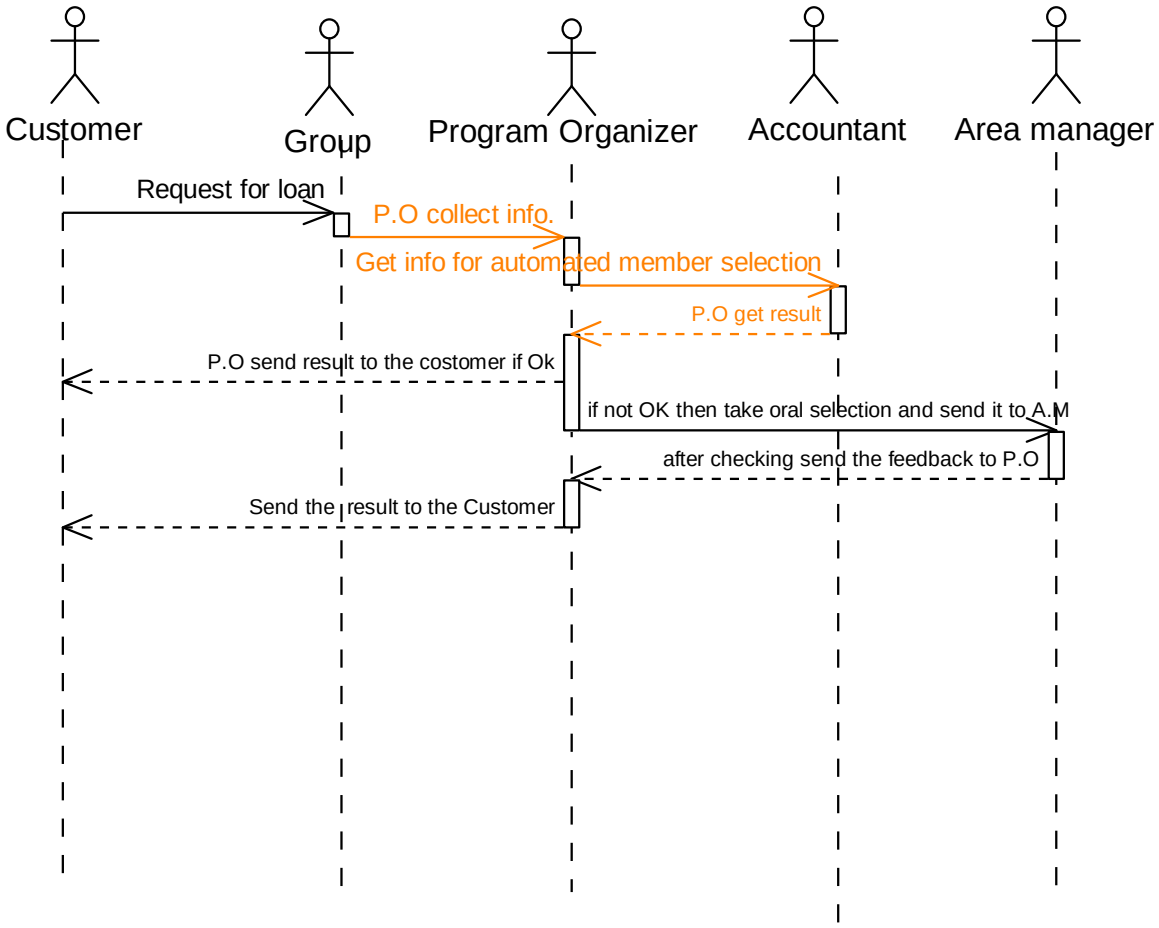


Fig. 5.4.3.1 Redefined sequence diagram for request for loan (Ref. Fig. 3.1)