Member Behavioral Model (MBM) for Members in Work Groups such as SHGs

A B Sagar
Hyderabad Central University,
Gachibowli, Hyderabad,
India

ABSTRACT
This paper presents the Member Behavioral Model [MBM] and Task Execution Cycle [TEC] for a member in a work group, typically SHG. The present idea of modeling the behavior of the member of the SHG greatly enhances the development of a holistic approach for the automation of SHG management. Every SHG [or any work group] has members. The behavior of such a member has not been modeled before. This paper attempts to model the member's behavior. The MBM is based on the functionality of members such that each member, after receiving a task, checks for its feasibility; and if finds it feasible, starts execution of the task. But if the task is infeasible, then assigns (forwards) the task to another member. When the task is assigned to others, the task goes into a sleep state at the assigner and after completion of the task, wakes up from sleep, and completes any remnant work, and reports to the supervisor. The MBM is modeled to accommodate additional features such as to allow the member for transparent and collaborative functionality. The modules and their functionalities are clearly defined in this paper. Implementation outline is also provided and various communication protocols are defined. This paper also defines & proposes a method for computing the performance index of a member.

Keywords
Member behavior model, task execution cycle, SHG formalization, framework for member.

1. INTRODUCTION
A SHG is a group of about 10 to 20 people who come together to form a small scale business, saving and credit organization. They pool their skills for their business growth personal development. SHGs also federate into larger organizations through clustering. At the cluster level, there are inter-group borrowings, exchange of ideas, sharing of costs and discussion of common interests. SHGs have been very effective in India. There were several thousands of them spread geographically all over the country. SHGs allow members to save money, provide employment, provide skill development. They help individuals to achieve their business and social objectives. They provide access to larger bank loans. *200 million* poor women today have access to savings and credit services through *3.36 million* SHGs all over the country. Over *30,000* branches of regulated banking structure are involved in this phenomenon and have mobilized loans of over *$ 450 million*. However, SHG management is not automated and only a few processes are computerized. Since majority of the members are either non-literate or semi-literate, it did not attract the industries to project them. Now, most of the processes in the SHG management are manual which makes SHG management stoo cumbersome. Automation of SHGs will greatly impact the management's performance and enhance the efficiency of the SHGs. At present, research on SHGs from computer science dimension exists only at piece-meal stage. There is no holistic approach for the SHG management yet. We believe that a holistic approach to the SHG management is very essential. The present idea of modeling the behavior of the member of the SHG[and even the SHG] and Task Execution Cycle, greatly enhance the development of a holistic approach for the automation of SHG management.

2. RELATED WORK
My previous paper, "People, SHGs and Social Objectives: A Formal Framework ", Special Issue of IJCCT, Vol-2, Issue-5, gives a brief overview of modeling SHGs. It provides the basis for the present paper and described the structure of SHGs, SHG members and the social values that are desired in the SHGs. The present paper proposes a model for the behavior of the member. Some of SHG processes were researched and papers were published regarding e-paper for writing payment details, book-keeping (register of minutes, register of accounts, cash book/ ledger/ vouchers/ receipts etc.), collection of information from remote rural clients, MISs, conducting financial transactions in remote rural areas, elimination/reduction of cash handling, e-purses, UIs for non-literate and semi-literate users for making epayments, smart cards with biometric technologies, etc.

3. MEMBER BEHAVIORAL MODEL
Every SHG [or any work group] has several members. The chief functionality of each member [or a SHG] is to perform certain tasks (business functions). Each member, after receiving a task, checks for its feasibility and if finds it feasible, then starts the execution of the task. But if the task is infeasible, then assigns (forwards) the task to another member (or SHG). When the task is assigned to others, the task goes into a sleep state at the assigner and after completion of the task, wakes up from sleep, and completes any remnant work, and reports to the supervisor. All the execution details will be stored in the Repst for log details.
The primitives of the MBM are explained below:

TMonitor: TMonitor is a continuously running process which keeps monitoring for items like tasks, acks, requests, reports, etc. When it receives an item it immediately forwards it to Rpt.

Rpt: Rpt's functionalities vary depending on the item it receives. If the received item is a task to be performed, firstly, it will notify the task sender through an ack about the receipt of the task and it will forward the task to ExTsk for execution. If the item is an ack, it will store the ack in the Repst against the corresponding task it has dispatched. If the item is a request, it will query the Repst and generate a report and dispatch it to the requester. If the item is a report, it will store the report in the Repst against the corresponding Request it has made.

ExTsk: The functionality of ExTsk is to log the stages a task passes through when a member (or a work group) executes the task. If the received task is feasible to that member (or work group), execution is started. Else, if the task is partially feasible, then task is split into three portions (feasible part, assign part, remainder part) and the assign part is

assigned to a different member/ work group. Soon after the report is received that the assign part is executed successfully, the remainder part is executed. After completing the execution of the whole task, i.e. all three parts, ExTsk sends the task completed notification to Rpt which is later forwarded to the notification rec ipients [task assigner, supervisor, etc]. While executing the task, ExTsk writes the intermediate task status to the store Repst through Rpt so that transparency is maintained regarding the status of task at every phase of execution of the task.

Repst: Repst is a repository and each activity along with all related information is stored in it. Whenever a request is received at the

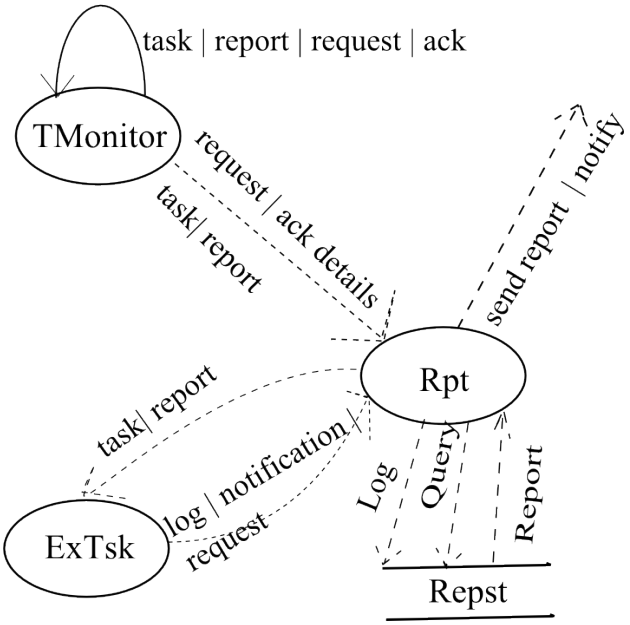

Fig 1: Member Behavioral Model

TMonitor, it is forwarded to Rpt which queries Repst and sends the query result as response. All logs are also stored in Repst.

Functionalities of Modules in MBM : In MBM, each module has different functionalities.

Table 1: Functionalities of Modules in MBM

Module	Actions	Item
TMonitor	Receives Task Receives Report Receives Ack Receives Request	<Task> <Report> <Ack> <Request>
Rpt	Receives Task Receives Report Receives Ack Receives Request Receives Log Receives Notification Sends Report Sends Notification Sends Query	<Task> <Report> <Ack> <Request> <Log> <Notification> <Report> <Notification> <Query>
ExTsk	Receives Task Receives Report Sends Log Sends Notification Sends Request	<Task> <Report> <Log> <Noitification> <Request>

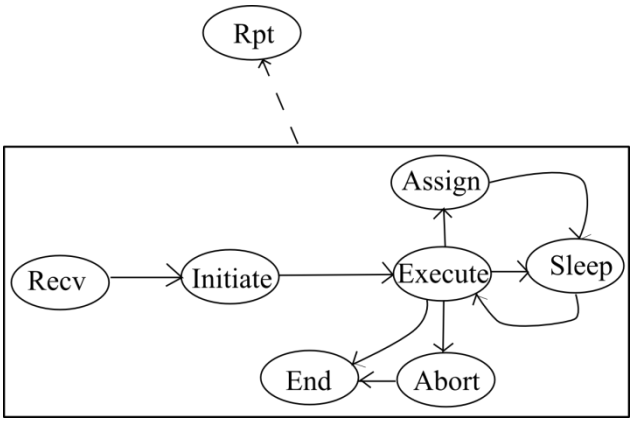

Fig 2: Task Execution Cycle [TEC]

4. TASK EXECUTION CYCLE

Each member (and also group's) responsibility consists of executing tasks. Each task has several properties, phases and associated financial and non-financial information.

The Task Execution Cycle (TEC) is the integral part of ExTsk of MBM. It has all the states that a task can undergo. The different states are described below:

Recv: The task is in Recvd state if it is just received. It will wait here until the member/ work group is ready to consider it for initiation and execution.

Initiate: The task moves to Initiated state if the member/ work group has considered execution.

Execute: The task moves to Executing state when the member/ work group has started execution or is continuing execution after waking up from sleep.

Assign: The task moves to assigned state if the task was assigned to a different member/ work group.

Sleep: The task moves to sleep state if the task was assigned to some other member or SHGs.

Abort: If the task was found to be infeasible or if the assignment fails for some other reason, then the task moves to aborted state.

End: The task reaches the final state by any means.

5. IMPLEMENTATION OUTLINE

Now, implementation of performance index in task execution can be done using reporting messages. Every activity/task during execution goes through certain states. Reporting messages are associated providing financial and non -financial information at each of the states. So, when the task reaches a certain state, a reporting message is to be sent to the notification recipient. The reporting messages carry both predefined information and dynamic information. Firstly, let us describe the task before we frame its execution cycle.

5.1 Task Properties

Executing a task is the basic operation of member and SHG. Every task, besides others, must have the following attributes:

Task_Id: Identification token of the task

Parent_Task: If the task is subtask of a bigger task, then this is the id of the parent task.

IsPartitionable: Yes, if the task can be split and distributed; No, if it has to be executed at only one place.

Risks: Risks involved with the task.

Task_Type: Whether the given task is technical or non-technical.

Creation_Date: Date when the task was created.

Excluded_Executors: Restricted executors list.

Task_Stakeholders: All the stakeholders who are related to the task.

Task_Status: Status of the task, indicating how far the task was completed.

Skills_Needed: List of skills needed to execute the task.

Skills_Levels_Needed: Level of skills needed for executing the task; excellent, fair, medium, poor skill levels.

Task_Infrastructure: Infrastructure that would be required for the execution of the task.

Task_Priority: Priority of the task; Low, medium, high.

Start_By_Date: The date by which one has to start the execution of the task.

Interdependencies: Interdependencies of the task.

Task_Supervisor(s): The member / SHG in-charge of the task.

Complete_By_Date: The date by which one has to complete the execution of the task.

Notification_Recipients: Stakeholders that are to be notified about the status of the task.

Responsibilites: Responsibilites that are to be undertaken by the task executor.

Task_ Initiator: Identity of the one who initiated the task.

Potential_Executors: Executors for whom the task is meant.

Cost_of_Task: Cost of the task estimated using variables like the resources and skills it involves and its significance (priority)

5.2 Reporting Messages

As discussed earlier, reporting regarding financial and non-finanicial information can be done using reporting messages associated with the . These reporting messages are application specific. However, some of the folloing reporting messages are mandatory.

Ack	AbortedTask
RecvdTask	EndedTask
InitiatedTask	RecvdRequest
ExecutingTask	RecvdReport
AssignedTask	

5.3 Description of Messages:

<Ack> - Ack to be sent to the report sender to acknowledge the receipt of the report.

<RecvdTask> - Message to be sent to the task sender acknowledging that the task was received.

<InitiatedTask>- Message to be sent to the notification recipients that the task execution is initiated.

<ExecutingTask> - Message to be sent to the notification recipients that the task is executing and its status.

<AssignedTask>- Message to be sent to the notification recipients that the given task was assigned to another member/ SHG

<AbortedTask> - Message to be sent to the notification recipients that the task execution was Aborted.

<EndedTask> - Message to be sent to the notification recipients that the given task execution has ended.

<RecvdRequest> - Message to be sent to the requester that the request was received

<RecvdReport> - Ack sent to the report sender that the report was received successfully.

Other operations related to tasks are sending reports, ack, requests, notifications,etc. When a request is received, it has to be processed and response (report) must be sent to the requester. When a task is received, an acknowledgment must be sent to the sender.

5.4 Formats of Reporting Messages*:*

<RecvdTask> - <RecvdTask> <taskId> from <task assigner Id> on <date> <time> <\RecvdTask>

<InitiatedTask> - <InitiatedTask> <taskId> on <date> <time> with <responsibilities> and <interdependencies> <\InitiatedTask>

<ExecutingTask> - <ExecutingTask> <taskId> <date> <time> <status> <problems> <\ExecutingTask>

<AssignedTask> - <AssignedTask> <taskId> <assignedto> on <date> <time> <\AssignedTask>

<SleepTask> - <SleepTask> <taskID> <wake up on> <event> <\SleepTask>

<AbortedTask> - <AbortedTask> <task Id> aborted due to <reason> on <date> <time> <\AbortedTask>

<EndedTask > - <EndedTask> <taskId> on <date> <time> <ended with success/failure/abort> <feedback> <performance understanding> <\EndedTask>

<Ack> - <Ack> <ackId> on <date> <time> for <taskId/ requestId/ reportId> from <memberId> <shgId> <\Ack>

<RecvdRequest> - <RecvdRequest> from <member Id> <shg Id> regarding <task Id> on <date> <time> <to be responded by date & time > <\RecvdRequest >

<RecvdReport> - <RecvdReport> <ReportId> on <date> <time> from <member Id> <shgId> for <taskId> and <requestId> sent on <date> <time> <\RecvdReport>

<Log> - <Log> <taskId> <taskStatus> <problems> <other info> <date> <time> by <member ID> <shg Id> <\Log>

5.5 Databases

During the implementation of transparency using MBM and TEC, we will be needed to maintain the following databases.

TASKS_DB, RECVD_TASKS_DB, ASSIGNED_TASKS_DB, SLEEPING_TASKS_DB, ABORTED_TASKS_DB, ENDED_ TASKS_ DB, INITIATED_TASKS_DB, ACK_DB, REQUESTS_DB, REPORTS_DB, TMONITOR_DB, EXTSK_DB, RPT_DB, LOG_DB

TASKS_DB : contains table which has information about every task.

TASKS_DB (Task Id, Task Type, Task Status, Task Priority, Task Supervisor, Task Initiator, Parent Task, Creation Date&Time, Activation Date & Time, Start By Date & Time, Complete By Date & Time, Potential Executors, IsPartitionable, Excluded Executors, Skills Needed, Skills Levels Needed, Notification Recipients, Cost of Task, Risks, Task Stakeholders, Task Infrastructure, Interdependencies, Responsibilites, Notification Recipients)

RECVD_TASKS_DB: Contains information about all received tasks.

RECVD_TASKS_DB(Task Id, Received From, On Date & Time, Start Date & Time, Completion Date & Time, Notification Recipients, Potential Executors, Infrastructure , Requirements, Skill Requirements, Responsibilities, Inter -dependencies , Risks, Priority , Parent Task)

INITIATED_TASKS_DB : Contains information about all initiated tasks.

INITIATED_TASKS_DB(TaskId, Initiation Time, Initiation Date, Notification Recipients, Initiating Member Id, Proposed Executors, Proposed Completion Date & Time , Skills Available For Task , Infrastructure Available , Priority Assigned)

EXECUTING_TASKS_DB: Contains information about all executing tasks.

EXECUTING_TASKS_DB(Task Id , Status , Problems , Notification Recipients,Members Involved , Skills Involved)

ASSIGNED_TASKS_DB: Contains info about assigned Tasks

ASSIGNED_TASKS_DB(TaskId, Assigned To, On Date & Time, Start Date & Time, Completion Date & Time , Notification Recipients, Infrastructure Requirements, Skill, Requirements , Responsibilities , Interdependencies , Risks , Priority , ParentTask)

SLEEPING_TASKS_DB: Contains info about sleeping Tasks

SLEEPING_TASKS_DB(TaskId, Slept On Date & Time, Waking Up Event , Woke Up Date&Time , Notification Recipients)

ABORTED_TASKS_DB: Contains info about aborted Tasks

ABORTED_TASKS_DB(Task Id, Aborted On Date & Time, Reason, Notification Recipients, Aborted By Member Id)

ENDED_TASKS_DB: Contains info about ended Tasks

ENDED_TASKS_DB(TaskId, Notification Recipients, Ended On Date & Time, Feedback, Performance Understanding)

ACK_DB: Contains info about acknowledgments.

ACK_DB(AckId, Notification Recipients, Sent On Date & Time, Corresponding Id of Task/Request/Report, Success / Failure, RecvdSenderDetails, PreparingAck, SendingAck , AbortingAck , EndedAckSend)

REQUESTS_DB: Contains info about requests.

REQUESTS_ DB(RequestId, Notification Recipients, Sent On Date & Time, Corresponding Task, Success/Failure, PreparingRequest, SendingRequest, AbortedRequest EndedRquesting)

REPORTS_DB : Contains info about reports.

REPORTS_DB (Report Id , Notification Recipients , Sent On Date&Time , Corresponding Task , Success / Failure , QueryingRepst GeneratingReport SendingRequestAbort/Fail EndReporting)

LOG_DB: Contains info about logs.

LOG_DB (Log Id , Made By , Date&Time , Log Data , Notification Recipients)

TMONITOR_DB: Contains info about reports.

TMONITOR_DB(Received Item, Sender's Id, Date & Time)

EXTSK_DB: Contains info about ExTsk

EXTSK_DB(Received Item , Senders Id , Date & Time , Sending Log/ Notification/ Request Id , Date & Time)

RPT DB: Contains info about Rpt.

RPT_DB (Received Item , Senders Id , Date&Time , Sending Report / Task / Query / Log / Notification Id Date & Time)

During the execution of the task, task status is to be notified to the notification recipients and requesters. Also, if a log or report is received, it is stored in the repository.

6. PERFORMANCE INDEX COMPUTATION

flow(info,SHG,S) is a function to indicate the information of task ('info') flowing from SHG to Stakeholders ('S'). In half-duplex communication system, the info flow is only unidirectional i.e. SHG to S. So, flow(info,SHG,S) holds true. But the converse, flow(info,S,SHG), is not true. That is, flow of info from S to SHG does not happen. There is no possibility for the stakeholders to request and obtain information of their choice in half-duplex mode. They only get what the SHG chooses to reveal. This feature provides for only a very limited transparency, also called selective transparency, because the SHG reveals only the information of its choice.

Since the stakeholders cannot request and obtain information of their choice, the governing body or the funding agency prescribes the 'info' that is to be revealed in each busi-

ness. Every SHG doing that business must disclose that prescribed 'info'.

For start up, we can begin with computation and comparison of static transparencies [transparencies in half-duplex mode of communication] of two SHGs doing similar businesses. Let us define flowrate as the rate offlow of information from SHG to S. *Since this is static transparency, the flowrate from S to SHG is zero*. Flowrate has two important dimensions namely Quantity of information and Quality of information.

For example, if a business function occurred 12 times, but it was reported only 6 times, then the flowrate is: number of business functions reported/ total number of occurrences.

Therefore, flowrate = 6/12 which is equal to 0.5 .

If the business function P_i was reported p_i times, with respect to N_i total occurrences , then its flowrate is f_i .

$$flowrate, f_i = \frac{p_i}{N_i}$$

The *Qualitative dimension* of flowrate associates a 'v' which is 'value'(or rank/priority) to each business function. This value will be assigned by the task initiators (funding / governing agency). Some business functions get a higher 'v' while other business functions get a lower 'v' based on their importance in that project.

Suppose a SHG has 'n' business functions P_1 , P_2 , P_3 ...P_n which are disclosed p_1 , p_2 , p_3 ...p_n respectively, and values v_1 , v_2 , v_3 ...v_n respectively, then the Static Transparency of that SHG can be derived as follows:

For example, let us consider two SHGs, G_1 and G_2 , each having five business functions P_1 , P_2 , P_3 , P_4 and P_5. Let the total occurrences of each business function be 12.

$$_{\text{static}}T_{\text{SHG}} = \sum_{i=1}^{n} v_i . \frac{p_i}{N_i}$$

$$we\ have\ flowrate,\ f_j = \frac{p_j}{N_j}$$

$$\therefore\ _{\text{static}}T_{\text{SHG}} = \sum_{i=1}^{n} v_i . f_i$$

$$\begin{aligned} _{\text{static}}T_{\text{SHG}} &= \sum_{i=1}^{5} v_i . f_i \\ &= (20 * 0.5) + (20 * 0.3) + (20 * 0.25) \\ &+ (20 * 0.5) + (20 * 0.16) \\ &= 10 + 6 + 5 + 10 + 3.2 \\ &= 34.2 \end{aligned}$$

Suppose that G_1 discloses all the messages. So G_1's messages will get the following disclosure values. $p_1 = 12$, $p_2 = 12$, $p_3 = 12$, $p_4 = 12$ and $p_5 = 12$. Let values (v_1 , v_2 , v_3 , v_4 , v_5) of messages be 20,20,20,20,20 respectively.

$$\begin{aligned} Now, flowrate f_1 &= p_1/(TotalOccurrences) \\ &= 12/12 \\ &= 1 \\ similarly, f_2 &= 12/12 = 1 \\ f_3 &= 12/12 = 1 \\ f_4 &= 12/12 = 1 \\ f_5 &= 12/12 = 1 \end{aligned}$$

$$\begin{aligned} _{\text{static}}T_{\text{SHG}} &= \sum_{i=1}^{5} v_i . f_i \\ &= (20 * 1) + (20 * 1) + (20 * 1) \\ &+ (20 * 1) + (20 * 1) \\ &= 100 \end{aligned}$$

Since the value is 100, G_1 is said to have good performance index.

Now G_2 :

Suppose that G_2 does not disclose all the messages. So G_2 's messages will get the following values. $p_1 = 6$, $p_2 = 4$, $p_3 = 3$, $p_4 = 6$ and $p_5 = 2$. Let values (v_1 , v_2 , v_3 , v_4 , v_5) of messages be 20,20,20,20,20.

$$\begin{aligned} Now,\ flowrate\ of\ P1, \\ f1 &= p1\ /\ (Total\ Occurrences) \\ &= 6/12 = 0.5 \\ similarly, f2 &= 4/12 = 0.33 \\ f3 &= 3/12 = 0.25 \\ f4 &= 6/12 = 0.5 \\ f5 &= 2/12 = 0.16 \end{aligned}$$

Since the value is 34.6, G_2 is said to have less performance index.

7. CONCLUSION

The present paper has successfully modeled a task and task execution of a work group such as SHG. The execution cycle of task execution is also generalized. This paper also calculates performance index of a task executor which can be either a member or a SHG. This paper presented a rudimentary framework for a member's behavior. This framework can be greatly useful in further research related to SHG automation. The future work would be modeling the SHGs, modeling the tasks, and modeling the collaborations among the SHGs so that effective federations can be formed.

8. ACKNOWLEDGMENT

I would like to thank Prof. Hrushikesha Mohanty, DCIS, HCU for his valuable guidance and thoughts for his guidance in my research work. This work is funded by Rajiv Gandhi National Fellowship.

9. REFERENCES

[1] People, SHGs and Social Objectives: A Formal Framework, A B Sagar, Hrushikesha Mohanty, Special Issue of IJCCT, Vol-2, Issue-5

[2] SHG Federations in India: A Status Report , APMAS 2007

[3] EDA Rural Systems and APMAS, Self-help groups in India: A Study of Lights and Shades, 2006

[4], Impact Of Self Help Groups (Group Processes) On The Social/Empowerment, MYRADA.

[5] Status Of Women Members In Southern India, SHG-bank Linkage Programme, New Delhi, November 2002

[6] "Impact Of Self Help Groups (Group Processes) On The Social/Empowerment Status Of Women Members In Southern India", in SHG-bank Linkage Programme, MYRADA, New Delhi, Nov. 2002.

[7] Optimizing SHGs , APMAS October 2005

[8] Indiastat.com, available at http://www.indiastat.com/table/socialandwelfareschemes/27/physicalprogressundersampoornagrameenrozgaryojana-sgry/449702/446706/data.aspx.

[9] Rural Microfinance Service Delivery: Gaps, Inefficiencies and Emerging Solutions , Tapan Parikh, 2007

[10] Self Help Groups In India A Study of the Lights and Shade www.edarural.com/documents/SHG-Study/Executive-Summary.pdf

[11] Self-Help Groups: A Keystone of Microfinance in India Women empowerment & social security, CS Reddy, APMAS, Oct 2005.

[12] Bank Penetration and SHG-Bank Linkage Programme: A Critique, Pankaj Kumar and Ramesh Golait, Reserve Bank of India Occasional Papers Vol. 29, No.3, 2009

[13] Self-help groups in india - a catalyst for women economic empowerment and poverty eradication, 33 rd global conference of ICSW, tours (France)

[14] SHG Federations: Development Costs and Sustainability, ACCESS Development Services, 2010

[15] Study on SHG Federations - Challenges and Opportunities, Sunil Kumar, Centre for Microfinance Research, Bankers Institute of Rural Development, Lucknow, 2010