Collaborations In Self Help Groups - A Computer Science Perspective Study and Framework

(Dr. A B Sagar, Dept of Computers and Information Sciences, Hyderabad Central University, Hyderabad)

Introduction

A SHG is a group of 10 to 20 people who come together to form a small scale business. The people [hereafter referred to as 'members'] pool their skills for their business growth and personal development. SHGs provide the means to distribute nation's wealth and resources into the society. SHGs also provide several other advantages such as - providing platforms for the poor women to discuss and resolve their problems; helping members manage cash flow deficits (maintaining food intake and overcoming emergencies), leading to improvement in quality and productivity of their only capital/resource i.e. human capital/resource; helping members avoid money lenders, especially to meet food and health emergencies; helping members invest in asset creation, diversify their occupations, and improve their risk-bearing capacities; promoting leadership qualities among their members; fostering women, even from conservative communities and regions, to interact with outsiders, particularly officials, including men; and establishing the linkage between banks and marginalized citizens, especially the women.

SHGs have to improve the way they develop products. Since their finances and technical capabilities are limited, a collaborated product development is a good solution. SHGs must deal with the need of ever increasing capability and complexity of product lines by incorporating new technologies with constraints of limited time and budget. So more research is indeed essential to develop Collaborative Product Development [CPD] solutions and configuration of processes, people, tools, and structural arrangements for SHGs to achieve the CPD goals.

SHG organization's structure is a hierarchical structure. The governing/funding agency is on the top of the hierarchy followed by SHG Federation, SHG and members.

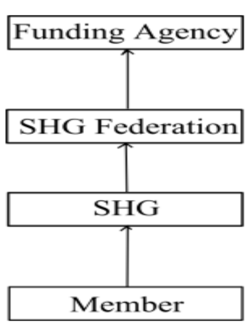
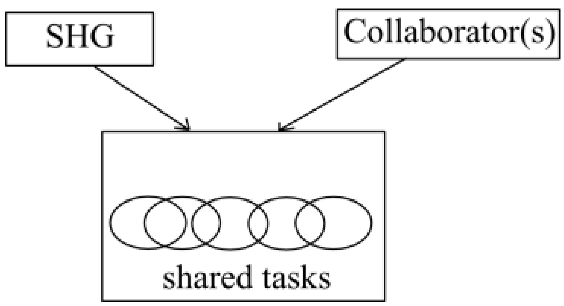
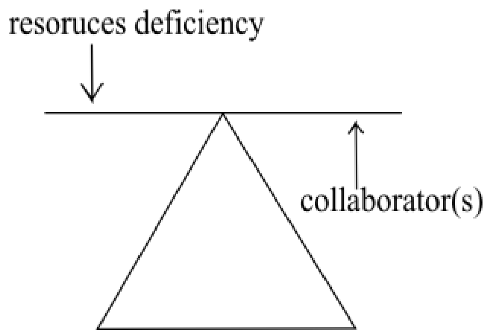


Fig. 1. SHG Management Hierarchy

The administration control flows from top to bottom, with funding agency generating the projects [tasks] and passes them to the lower levels for execution. Thus, members and SHGs receive tasks for execution. Sometimes SHGs receive tasks which they cannot complete with their existing resources [resources mean skills, materials, machinery, workers, etc]. In such cases, they need to collaborate with other SHGs who have the required resources. Collaboration is a joint effort of multiple individuals or SHGs to accomplish a task or project. Collaboration has more credibility, influence, and ability to accomplish objectives than a single entity. Collaboration with other SHGs provides several other benefits to SHGs such as - lowering production costs, access to less expensive labor, increased creativity and innovation, better products/services, improved revenue opportunity, shrinking distances and time, etc. Thus, collaborations are very beneficial to SHGs. But there are several issues such as communication needs, identity management, performance, mutual benefits, costs, etc. involved in collaborations. So, this paper addresses each of these issues.



(a) Shared Tasks



(b) Collaborators provide task requirements

Fig. 2. Collaborators

Collaboration is something more than mere coordination or cooperation. Understanding the difference between the words coordination cooperation and collaboration is challenging because they have often been used interchangeably. All these three words include, “working together” in their primary definition. However, each word differs slightly on who is working together, and what they might be working on. Merriam-Webster Online (2009) further defined these words as follows:

1. Coordination: the harmonious functioning of parts for effective results
2. Cooperation: to act together or in compliance for mutual benefit
3. Collaboration: to work together jointly, especially in an intellectual endeavor;

Collaboration involves identifying the right SHG with most relevant skills, personalities, knowledge, work-styles, and ethical values, ensuring they share commitment to the collaboration task at hand, and offering them environment, tools, knowledge, training, process and facilitation to guarantee they work together efficiently.

Related Work

Some of SHG processes have been under research 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. 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.

In the agent collaboration, Smith and Davis described agent collaboration as a pre-designed role within each agent logics to establish goals adaptation phenomena between agents, provided that there are no resource conflicts. However, the research defines agent collaboration as “the process of dynamically forming a team of agents toward the achievement of common goal” The team formation process has been designed in different approaches, including, agent motivation, execution plan, organization structure, built-in objective. For example, Joint Intention introduces shared beliefs. Another notion, Shared Plan, is based on sharing the execution plan. Planned Team Activity is based on individual BDI [Belief, Desire, Intention] and predefined plans within agent internal states. Wooldridge and Dunne, also presents a model based on the desire to achieve one of a set of goals. This set of goals is linked with the coalition choice then this choice leads to corresponding collaboration. There are also some attempts which rely totally on the capability of agent interactions using standard communication protocols. For example, Vieira et al. has been developed knowledge based semantic to be incorporated into agents programming language known as AgentSpeak. This semantic expand AgentSpeak logic to recognize, agent communication messages and transform them into knowledge and subsequent action related to this knowledge.

Now, there are already some achievements in agent collaboration. For example, Liu proposed a multitasking, multi-strategy and multi-round contract net considering all aspects, which can efficiently solve problems of task distribution and resource distribution. Chen proposed an agent collaboration mechanism based on agent’s belief commitment, in which designed result of agent is represented by “belief”. Collaboration between agents can be completed by belief commitment.

Scholar Liu applied auction theory in economics to contract net protocol and proposed an improved negotiatory scheme of contract net based on auction theory. In that scheme, task's dynamic allocation is realized by agent's free competition. In this way, getting and updating other agents' resources and abilities can be much easier. Also, it supports the dynamic variation of agent's knowledge and ability, so that, it can cut down the amount of communication among agents and reduce collaboration time. AlHashel applied astringency and stability to the protocol to improve the stability of agent's collaboration. But the realization of agent is too complicated. Jennings and Roda proposed acquaintance model according to the collaboration mechanism in real life.

The present paper proposes a framework for self help groups whose working environment can change dynamically. The concept of self help groups is formalized and the scheme adopts some organizational rules to establish self help group and design the communication and establishment mechanism.

Formalizing the SHG Collaborations

Collaborations occur over tasks. So, before formalizing the collaborations, let us formalize the tasks.

Tasks are business functions that are executed by all SHGs. Task execution is the basic functionality of SHGs and they are also the basis for collaborations. Once a task was given to a SHG, it is the responsibility of that SHG to execute it and return the result to the issuing authority. A task, in the present model, is an entity with several attributes. Every task, besides others, has the following key attributes: Task_Id, Parent_Task, IsPartitionable, Risks, Task_Type, Creation_Date, Excluded_Executors, Task_Stakeholders, Task_Status, Skills_Needed, Skills_Levels_Needed, Task_Infrastructure, Task_Priority, Interdependencies, Notification_Recipients, Task_Supervisor, Responsibilities, Task_Initiator, Potential_Executors, Cost_of_Task, Start_By_Date, Complete_By_Date, etc.

Each task [T] has a 'start state' and a 'final state'. Before beginning of execution of a task, the task is at 'start state'[S]. During execution, the task moves slowly towards completion, and finally reaches the final state called the goal state [G]. The distance between the task's start state and the goal state is D.

The subdistances d1, d2 , d3 , ... dn are the distances that a task moves towards goal G with the completion of each of its subtasks. Hence, for n subtasks, total distance D is:

$$D = \sum_{i=1}^{i=n} d_i$$

From Fig: 3, a simplified mathematical representation of a task would be:

T ask, T = (l_i , d_i , t_m , n) where l_i is the initial location of the task, d_i is the destination of the task, t m is the total subtasks and n is the number of collaborators involved. If a task has no subtasks or there exists a constraint that it should not be split into subtasks, then t = 0. And, if a task does not have any collaborators, then n = 0. A task is called a complex task if t > 1 and n > 1.

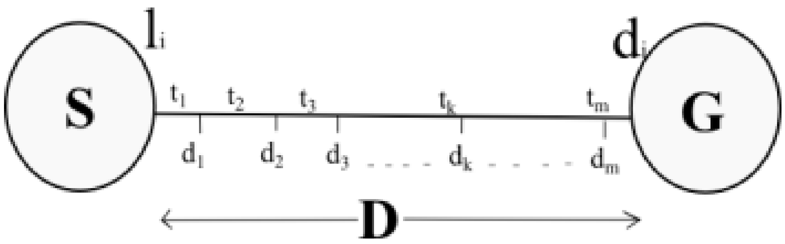


Fig. 3. Task - Goal - Distance

If the task has several subtasks, then with the completion of each subtask it moves a distance d_m. The distance moved after k subtasks is $\Sigma d_k - l_i$ and remaining remaining distance is obtained from the difference between d_k and Σd_m. When $\Sigma d_m = D$, the task execution is finished.

If complexity i is the complexity involved in executing the task t_i , that is, $complexity_i$ is the complexity to move the task a subdistance of d_i , then total complexity of the task, $Complexity_T$ is the summation of complexities of all subtasks. If there are n total subtasks, then

$$Complexity_T = \sum_{i=1}^{i=m} (t_i.complexity)$$

Collaborated Distance: A complex task has subtasks and collaborators share the the subtasks. The total distance moved by collaborators is called collaborated distance. The collaborated subtasks may be executed sequentially or nonsequentially. If there are n total subtasks and out of them k subtasks are collaborated subtasks, then, Collaborated Distance, CD = $\sum_{i=1}^{i=k} d_i.$

And, the complexity of collaborated subtasks is :

$complexity_{CD}$ = $\sum_{i=1}^{i=k}(t_i.complexity)$

Task-Collaboration Graph[Fig: 4] *This* graph gives the detail of the subtasks and the corresponding collaborations. This also details the concurrent subtasks. The subtasks that are enclosed between the vertical lines indicate concurrent subtasks. The weighed edges connecting the subtasks denote the subdistances i.e. the distance a task moves towards the destination when the task is completed. When multiple collaborators execute a subtask, then they all have arrows pointing them. Finally the task reaches the goal state.

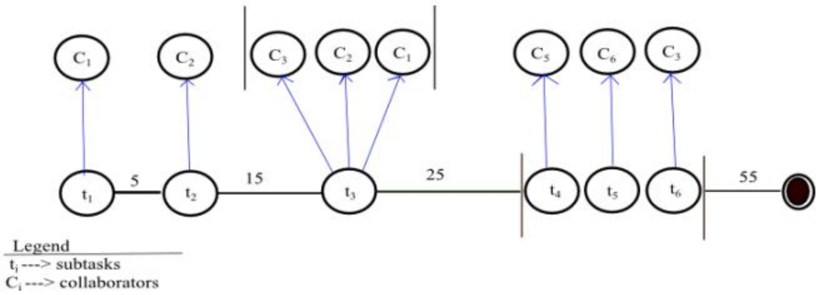


Fig. 4. Task Collaboration Graph

Task Classification: Task classification helps in defining the complexity of the task. A task is classified as independent if, for its execution and completion, it does not need any collaborators. If the task needs the collaborators then it is a collaborative task. And, if the task is completely infeasible for the SHG, and needs to be transferred to some other SHG, then it is an assignable task. The complexity of an independent task is zero, complexity of a collaborative task is some arbitrary value calculated depending on the task, complexity of assignable task is infinity. If the collaboration is needed for a short span of time, then it is 'partial collaboration', and if the collaboration was needed throughout the task, then it is 'complete collaboration'. Complexity of partial collaboration is lesser than the complexity of complete collaboration. If the collaborative tasks need both parties [i.e.both the task owner and the collaborator] to involve in the execution, then it is 'synchronous', else if part of the task is executed independently by the collaborator, then it is asynchronous. Complexity of the synchronous tasks is greater than the asynchronous tasks.

The spatial or geographic dimension is that the collaborators are either in the same place (co-located) or in different places (remote). Complexity in case of remote collaborators is greater compared to co-located collaborators.

Definition 1. Independent Task: An independent task is executed solely by the SHG to which it was designated. The SHG cannot involve others in the execution of this task. It could be that the task is confidential, or that it is designed to evaluate the performance of the SHGs.

Definition 2. Collaborative Task: A collaborative task can be executed with the help of collaborators. Usually, collaborative tasks are complex and require resources and skills which cannot be found in a single SHG. The SHG to which this collaborative task is designated, it should remain as one of the executor. It cannot completely transfer the task to another SHG.

Definition 3. Assignable Task: An assignable task is the one which can be executed by the designated SHG or transferred to some other capable SHG, but not to be collaborated. There may be additional constraints regarding what type or how far away SHG (co-located/ remote) may get this task.

Definition 4. Partial Collaboration: The partial attribute is when only a portion of the task uses collaborations or when the collaboration is for only a short period of time. In other words, if a task can be partitioned into ten subtasks, then 'partial' implies any number of subtasks less than ten.

Definition 5. Complete Collaboration: The complete attribute is when all portions of the task use collaborations. For example, if a task can be partitioned into ten subtasks, then 'complete' implies all the ten subtasks.

Definition 6. Synchronous Collaboration: Synchronous attribute is used when the collaborator works 'simultaneously' with the SHG.

Definition 7. Asynchronous Collaboration: Asynchronous attribute is when the collaborator does not work 'simultaneously' with the SHG. The working site may be collaborator's location or the SHG's location.

Definition 8. Co-located Collaborator: Co-located implies that the SHG and the collaborator are located at the same geographical location. The co-location can further be defined as to be located at a distance such that they do not need any transportation facilities to reach each other.

Definition 9. Remote Collaborator: 'Remote' implies that the SHG and the collaborator are not located in the same geographical region. Remoteness can further be defined as to be located at such a distance that they need transportation facilities to reach each other.

Definition 10. Partial Collaborative Task: Partial collaborative task is the one in which collaboration is used for only a limited extent. The partialness is related to time period and extent of task. When a task is referred in terms of time,say five months, then a partial collaborative task is the one which uses collaborations for only a month or two. And, When a task is referred in terms of extent,say production of five thousand baskets, then partial collaborative task is the one which uses collaboration for the production of a few hundred baskets.

Definition 11. Complete Collaborative Task: Complete collaborative task is the one in which collaboration is used totally, i.e. till the end of the task.

Definition 12. Synchronous Partial Collaborative Task: This task is a partial collaboration task and has an additional constraint. The 'synchronous' attribute necessitates that the collaborator works simultaneously with the SHG.

Definition 13. Asynchronous Partial Collaborative Task: This task is a partial collaboration task but that there is an additional attribute 'asynchronous'. The asynchronous attribute makes it flexible for the collaborator to need not work simultaneously with the SHG. The collaborator can be working in its own site and its own time schedules.

Definition 14. Synchronous Complete Collaborative Task: This task is a complete collaboration task and has an additional constraint 'synchronous'. The 'synchronous' attribute necessitates that the collaborator works simultaneously with the SHG.

Definition 15. Asynchronous Complete Collaborative Task: This task is a complete collaboration task but that there is an additional attribute 'asynchronous'. The asynchronous attribute makes it flexible for the collaborator to need not work simultaneously with the SHG. The collaborator can be working in its own site and its own time schedules.

Definition 16. Co-located Synchronous Partial Collaborative Task: This is a synchronous partial collaboration task but with the additional attribute of co-location. The attribute 'co-located' implies that the collaborator and the SHG are both geographically close to each other. The co-location can further be defined as 'being located at a distance such that they do not need any transportation facilities to reach each other'.

Definition 17. Co-located Synchronous Complete Collaborative Task: This task is similar to synchronous complete collaborative task but with an additional attribute of co-location. 'Co-located' implies that the collaborator and the SHG are located in the same geographical region. The co-location can further be defined as to be located at a distance such that they do not need any transportation facilities to reach each other.

Definition 18. Co-located Asynchronous Partial Collaborative Task: The task uses collaborations partially, works asynchronously and is located at the same geographical location.

Definition 19. Co-located Asynchronous Complete Collaborative Task: The task uses collaborations till the completion of the task, works asynchronously and is located at the same geographical location.

Definition 20. Remote Synchronous Partial Collaborative Task: This task differs from Co-located Synchronous Partial Collaborative Task in its attribute of remote. 'Co-located' implies that the collaborator and the SHG are located in the same geographical region where as 'remote' implies that they are not located in the same geographical region. Remoteness can further be defined as to be located at such a distance that they need transportation facilities to reach each other.

Definition 21. Remote Synchronous Complete Collaborative Task: This task is similar to synchronous complete collaborative task but with an additional attribute of co-location. 'Co-located' implies that the collaborator and the SHG are located in the same geographical region. The co-location can further be defined as to be located at a distance such that they do not need any transportation facilities to reach each other.

Definition 22. Remote Asynchronous Partial Collaborative Task: The task uses collaborations partially, works asynchronously and is located at the same geographical location.

Definition 23. Remote Asynchronous Complete Collaborative Task: The task uses collaborations till the completion of the task, works asynchronously and is located at the same geographical location.

Process Flow In A SHG:

SHG process flow consists of four main parts namely Collaborations Monitor & Communicator, Collaborations Processor, ExTsk and Resources Repository.

Collaborations Monitor & Communicator: It is the SHG's interface for collaborations. It receives collaboration requests and forwards them to Collaborations Processor. And also communicates with other SHGs regarding collaborations.

Collaborations Processor: All collaboration requests forwarded from the Collaboration Monitor are analyzed here. This module will communicate with ExTsk and Resources Repository and analyzes the future resources usage. If excess resources are available than needed for the execution of present tasks, then it will indicate the Collaboration Monitor that it is available for collaborations.

ExTsk : This is the module logging the status of the running tasks.

Resources Repository: All the information regarding SHG's resources are maintained here. It holds information of the resources currently being used, available resources, and resources in the network that are available through collaborations.

When SHGs receive tasks of such size and complexity that no single member or SHG itself is capable of effectively executing them, then they resort to collaborations. A collaboration can be defined as an initiative taken by a SHG to move the task from its start state (S) to (or towards) the goal state (G) by joint efforts with other SHGs. A ollaborator is the SHG which moves [or helps in moving] the task [or subtask] towards the goal (G). A subdistance d i corresponds to a subtask t i [from Fig. 3]. Since the subdistances vary in length with each other, it is more appropriate to measure the progress of the task not by the number of completed subtasks, but by the total distance it moved towards the goal. With addition of collaborators, the goal can be reached more quickly. That is, with the addition of collaborators the total distance D is covered quickly.

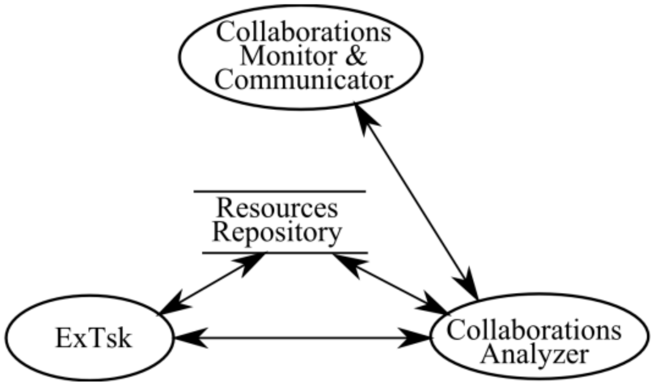
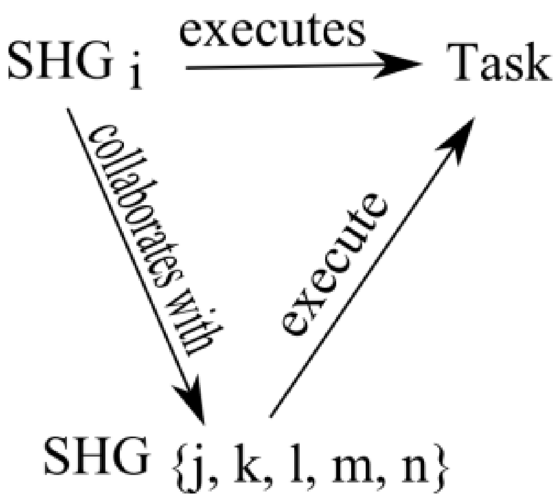


Fig. 6. SHG Process Diagram

Let S be the universal set of SHGs in the network.

$$S = \cup \left\{ SHG_i \right\}_{i=1}^{n}$$

A SHG SHG i might collaborate with one or more SHGs for executing a task T. If SHG_i collaborates with the SHGs SHG_j , SHG_k , SHG_l , SHG_m , ...SHG_n for executing a task T , then the collaboration is given as : collab(SHG_i , $SHG_{\{j,k,l,m,n\}}$, T).



Publish-Subscribe Model

The Publish-Subscribe Model is the process of collaboration in SHGs. It executes at every SHG internally. The model considers two basic methods for finding a collaborator: (1) Advertising collaboration specifications and then waiting for collaborators to subscribe (2) Searching the SHGs in the network and finding out suitable collaborators.

Collaboration specification contains resources requirements, collaboration goals, collaboration type, collaboration duration, etc. Collaboration goals are center constructs of the collaboration specification and are fundamental for every collaboration. Resources requirements imply capabilities and process specifications. Capability specification gives the details of the capabilities that a prospective collaborator should possess and process specification details how the task is to be executed i.e. plan of execution of the task. The overlapping between the resources requirement of the advertising SHG, and the excess resources available with another SHG, is the initial factor for the latter SHG to subscribe as a prospective collaborator to the former SHG.

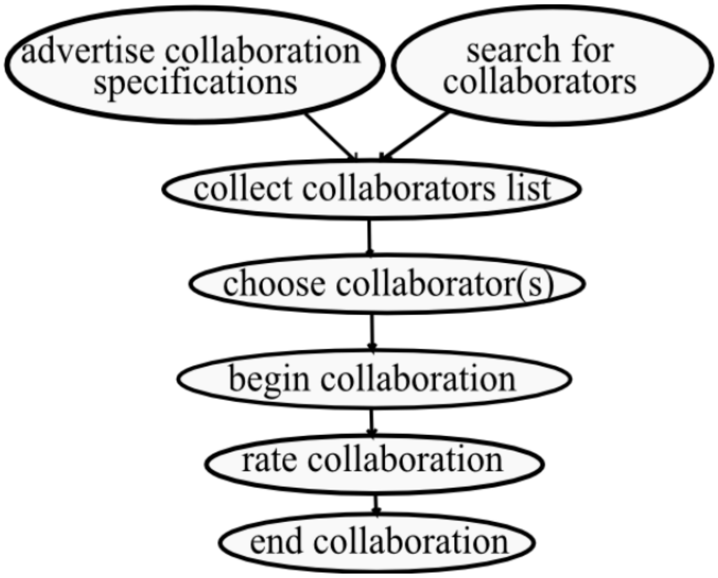


Fig. 7. Publish-Subscribe Model

$$max(overlap(SHG_i.Task.required\_resources, SHG_j.resources\_available)) \implies collaborator(SHG_j, SHG_i)$$

Inviting SHGs for collaboration is done by propagation of a collaboration specifications advertisement called collab_advt. The format of the collab_advt is given in Fig: 8.

shg_id
task_id
task description
resources requirement
capability specification
process specification
collab_duration

Fig. 8. Collaboration Advertisement [collab_advt]

Two basic functions are required for establishing collaboration: SEND function to send a collab_advt or response, and RECEIVE function to receive a collab_advt or response. Every SHG will have these SEND and RECEIVE functions. For example, a SHG ‘A’ sends a collab_advt by SEND(X), another SHG ‘B’ receives it by RECEIVE(X). After analyzing the collab_advt X, ‘B’ sends the response using SEND(X-Response) and ‘A’ receives the response using RECEIVE(X-Response).

The SEND/RECEIVE pair are enough for collaboration message passing. Two more functions are also used, STOP and ACK. The STOP function stops advertising. It marks the receipt of enough collaborators' responses. ACK function is for acknowledging the receipt of response from a responding SHG.

Table 1. Collaboration Process

Publisher	Collaborator
1. Publish the collab_advt	
	2. Receive collab_advt
	3. Analyze the collab_advt
	4. Respond
5. If collaborator is found, start collaboration	

Table 2. Constructs of the Communication Language

Construct	Description
Create_Collab_advt	SHG creates an advertisement of collaboration over a task
Broadcast	Send the collab_advt over the SHG network
Available	A SHG notifies other SHGs about its availability for collaboration
Send	The sender sends the collab_advt or response to a specific SHG
Receive	The receiver receives the collab_advt or response
Response	The receiver responds to the collab_advt
Establish_Collab	The sender and receiver agree for collaboration
Signal	The sender signals the receiver that it can begin the execution of the task
Notify	The receiver or sender notify each other of some information
Query	The receiver queries the sender about some information it does not know about the task
Inform	The sender sends some information to the receiver
CannotDoTask	The receiver refuses to perform the task stating the reason for the refusal
Alert	The sender alerts the receiver about some event the receiver needs to react to. An alert message is considered urgent
Subscribe	The sender requests the receiver to inform the sender whenever the status of the receiver or task changes
Cancel	The sender informs the receiver to cancel a previously established subscription

Table 3. Communication Messages

Sender	Receiver	Construct	Subject	Data
Any SHG	Any SHG	Broadcast	Collaboration advertisement broadcast	collab_advt
Any SHG	Any SHG	Available	Available for collaboration	resources details, availability details
Any SHG	Any SHG	Receive	Receiving collaboration advertisement /response	collab_advt / agreement details, availability details, resources details
SHG(A)	SHG(B)	Send	Sending collaboration advertisement/ response	collab_advt / agreement details, availability details, resources details
SHG(A)	SHG(B)	Signal	Begin task execution	task_id, date and time details
SHG(A|B)	SHG(A|B)	Notify	Notification	notification data
SHG(B)	SHG(A)	Query	Query	query regarding the task, or other issues
SHG(A)	SHG(B)	Inform	Information	information required by the query
SHG(B)	SHG(A)	CannotDoTask	Cannot do task	reason for refusal of the task, task_id
SHG(A)	SHG(B)	Alert	Urgent	alert message
SHG(A)	SHG(B)	Subscribe	Subscribe	subscription details, notification details
SHG(A)	SHG(B)	Cancel	Cancel subscription	reason for canceling subscription

Search For Collaborator:

Functionally, each SHG is a pool of resources. What makes a SHG a collaborator is the connectedness of the SHG to the task's need. Hence, finding a suitable collaborator involves finding SHG [or SHGs] with most suitable resources for the task's need and located at the shortest distance. The problem of finding suitable collaborator(s) now becomes a goal reaching problem through shortest distance. Given a task (infeasible), we need to design a scale-effective cluster of SHGs for solving the problem. We have to search through the entire SHG network and identify the right 'combination' of collaborators who can optimally execute the task. This search process can be optimized by limiting the search to a smaller subspace of 'good' SHGs. This subspace is formed by moving/promoting well performing SHGs into the subspace. After the best collaborators are chosen, task execution begins based on the task execution plan.

Task Execution Plan

Task_Id: T3032011(5)
Task_Name: Frames Construction

Subtasks	Collab-orator_Id	Collabora-tion Details	Start_Date	Finish_Date
Bending Rods	SHG-20909	asynchonous collaboration	july 4, 2011	aug 3, 2011
Making Frames	SHG-23233	synchrnouns collaboration	aug 5, 2011	aug 21, 2011
Painting	SHG-98970	asynchronous collaboration	aug 22, 2011	sep 3, 2011

Fig. 9. Task Execution Plan

Collaboration Rating:

Collaboration rating is the penultimate process in the Publish Subscribe Model. When the SHG advertises for collaborators, it receives acceptance from some SHGs that are interested in collaborations, then the SHG needs to make a decision and choose the best collaborator. This rating aids in choosing the collaborators. To create this rating for each SHG, the Publish Subscribe model has this 'collaboration rating' step. That is, at the end of each collaboration, the SHG and collaborator rate each other. Thus, every SHG that has participated in a collaboration will obtain a rating. Though several criteria are considered for rating the collaboration, satisfaction degree and trust degree are considered important. SHGs with good collaboration ratings form a subgroup of prioritized collaborators. A certainty value 'W' is derived to compute the satisfaction degree. To compute satisfaction degree of a SHG over a task, multiply the certainty value with a variable factor $\lambda \mid \lambda \in (0,1)$.

$$SatDegree(SHG\_ID, Task\_Id) = \lambda * W$$

If the collaborating SHG is satisfied with the collaboration quality, λ is positive, if not satisfied, then λ is zero. The certainty value 'W' is computed from several characteristics of the collaborator and they are given in the table below.

Characteristics of the collaborator	Value
Commitment of collaborator towards the task execution	4
Open-mindedness of the collaborator	3
Timeliness of the collaborator in sending information	5
Willingness to continue even during difficult situations	6
Perceptive capability of the collaborator	3
Skilfulness at giving/ receiving even negative responses	4
Self-managing and requires "less supervision"	3
Thinking differently and bringing new perspectives	5
Comfortability with ambiguity	4
Adaptiveness to changes/new requirements	5
Significance of stake in the outcome of collaboration	4
Ability of the collaborator in improvising performance	3
Sense of urgency about collaboration	5
Encouragement of equal participation among all members	3
Optimism and being on the upbeat	4
Pragmatism and knowing how ideas get executed	5
Collaborator has lots of ideas	3
Collaborator thinks strategically	5
Collaborator has good knowledge required for the task	4
Collaborator has good connections and networks	3
Collaborator is skilful at helping the group reach consensus	5
Collaborator is well organized	3
Collaborator is felt trustworthy	4
Collaborator has experience being a collaborator	3
Collaborator is gregarious and dynamic	4
Total	100

Conclusions and Perspectives

This paper formalizes a collaborations in SHGs. Tasks are also formalized and attributes are briefly defined. Terms such as goal, distance, subdistance, subtasks, collaborated distance, task-collaborator graph, and task classification are introduced and defined as per the scope of SHGs. Publish Subscribe Model is proposed and the collaboration process is described. Several constructs of communication language and various communication messages are also defined. The process of searching for a suitable collaborator was given. Also, collaboration rating is defined and the process of computation of collaboration rating is also described.

Acknowledgments

The author is grateful to Prof H Mohanty [DCIS, HCU] for pointing out the research direction explored in this paper as well as for helping in framing and presenting the theory presented here. This work is funded by Rajiv Gandhi National Fellowship (RGNF), India.

References

1. People, SHGs and Social Values: A Formal Framework, A B Sagar, H Mohanty, Special Issue of IJCCT, Vol-2, Issue-5, 2011.

2. Collaborative Product Development in SMEs: Requirements and a Proposed Solution Xiaozhen Mi, Weiming Shen, Wenzhong Zhao, The 9th International Conference on Computer Supported Cooperative Work in Design Proceedings, 24 ˝U26 May 2005, Dalian, China, Vol. 2, 876 ˝U882.

3. A Conceptual Design of an Adaptive and Collaborative e-Work Environment, Obinna Anya, Hissam Tawfik and Atulya Nagar, Proceedings of the First Asia International Conference on Modelling & Simulation, (AMS'07).

4. X. Mi, and W. Shen," Computer supported Collaborative Product Development: a Review", Proceedings of CSCWD2005, Coventry, UK, 2005.

5. On Key Issues in Information System for Collaborative Product Development Process, Weigang Li, Wenbin Wang, Junyi Shen, Proceedings of the 10th International Conference on Computer Supported Cooperative Work in Design, 2006 IEEE.

6. PEACE-VO: A Secure Policy-Enabled Collaboration Framework for Virtual Organizations, Jianxin Li, Jinpeng Huai, Chunming Hu , 26th IEEE International Symposium on Reliable Distributed Systems 1060-9857/07, 2007

7. Synchronous versus Asynchronous Collaboration in Situated Multi-agent Systems Danny Weyns, Tom Holvoet, AAMAS'03, July 14 - 18, 2003, ACM.

8. Collaboration rules for autonomous software agents,Sarosh N. Talukdar, Decision Support Systems, (1999), p269 - 278.

9. SHG Statistics http://www.indiastat.com/table/socialandwelfareschemes/27/physicalprogressundersampoornagrameenrozgaryojanasgry/449702/446706/data.aspx