Proposing an Architecture for Collaborative Product Development in Self Help Groups

A B Sagar

Dept of Computers and Information Sciences, Hyderabad Central University, Hyderabad, India

bablusagar@gmail.com

ABSTRACT

A Self Help Group (SHG) is a group of 10 to 20 people who come together and pool their resources to form a small business. SHGs also evolve into larger organizations through federations. SHGs have been very effective in India. There were several thousand of them spread geographically all over the country. Automation of SHGs will greatly impact the SHGs’ performance in business and in society. SHGs are gaining importance in research studies, and hence formalizing the SHGs is a fundamental initiative for further research. Sometimes SHGs receive tasks which they cannot complete with their existing resources. In such situations, they collaborate with other SHGs who have the required resources. SHGs also collaborate to achieve higher objectives (goals). There is no proposed framework for Self Help Groups, and Collaborations among Self Help Groups. Objective of this study is to provide a formal framework for SHGs and also model a rudimentary framework for collaborations among SHGs. Since the primary functionality of SHGs is to execute tasks, and also tasks are the basic reason for collaborations. So, to develop a framework for collaborations, we first need to formalize a task. We also formalize a member and SHG, and then proceed to develop framework for collaborations among SHGs. This paper proposes a framework for collaborations in SHGs and also proposes a metric of collaboration. A rudimentary framework for collaborations among SHGs is obtained; and it is also observed that, through further research, holistic models can be developed based on this work.

Keywords: SHG, Self Help Group, Collaboration, social values, ethics, Collaborativeness, Collaborative Product Development, CPD

1. 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 cashflow 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 fidels, 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.

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.

![SHG Management Hierarchy](image1)

**Fig. 1: SHG Management Hierarchy**

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](image2)

**Fig. 2: Collaboration**

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.

2. 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 payments, 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.

3. ORGANIZATION OF THE PAPER

This paper begins with modeling SHG entities. Next it models a SHG and specifies its collaboration rules. Next it models a task and its attributes are briefly defined. Terms such as goal, distance, subdistance, subtasks, collaborated distance, task-collaborator graph, and task classification are defined with respect to SHGs. Publish Subscribe Model is proposed and the collaboration process is described. Several constructs of communication language and various communication messages are described. Next, the process of searching for a suitable collaborator is given. Next, the process of computation of collaboration value is described. Collaboration architecture is proposed. Finally, conclusion and perspectives are given.

4. MODELING SHG ENTITIES

The basic SHG entities are member, task and collaborative. Before formalizing the entities, let us make some basic assumptions regarding SHGs.

1) SHGs can be viewed as a pool of resources.
2) SHGs need resources to complete the business tasks.
3) Resources of SHGs are unevenly distributed.
4) Some SHGs possess excess resources through which they not only can execute their own tasks, but can also execute other SHGs’ tasks while some SHGs have a very few resources such that they are need collaborators to execute their tasks.
5) SHGs are basically collaborative.
6) Collaboration is a mutually beneficial activity and it offers the involved parties - environment, tools, knowledge, training, process and facilitation to guarantee they work together efficiently.

4.1 Formalizing a Member

A SHG member can be formalized as m = (mID, Skills, Tasks, COLLABS). ‘mID’ is the unique identifier for identifying a member of a SHG, ‘Skills’ are the talents that are possessed by the member, for example: carpentry, plumbing, painting, electrical work, etc; ‘Tasks’ are the business functions that a member can perform, ‘COLLABS’ represents the collaborations that a member has with other members or SHGs.

1. \( \forall m \in M, \exists mbf \mid \text{has}(m, mbf) \land | mbf | \neq 0 \)

Every member has specific business functions.

2. \( \forall mbf, \text{feasible}(m, mbf) \Rightarrow \text{has}(m, mbs) \lor \text{training}(m, mbs) \)

If a member’s business functions are feasible, it means that he has the required business skills or he obtains the skill through training.

3. \( \text{achievable}(m, mbg) \Rightarrow \forall mbf, \text{feasible}(m, mbf) \)

A member’s business goals are achievable if all his business functions are feasible.

4. Member task execution rule:

if \( m(\text{task}, \text{feasible}) \) then \( \text{execute}(m, \text{task}) \) else \( \text{find}(m, \text{collaborators}, \text{task}) \)

After a member receives a task, he will check if he can execute the task with his skills. If the task is feasible to him, then he executes it, else he tries to find collaborator(s) to execute the task.

5. Member collaborations forming rule:

If \( \text{match}(m_i, \text{Task.needed.resources}, m_i, \text{skills}) \land (\text{negotiations}(m_i, m_j) = \text{success}), \Rightarrow \text{collab}(m_i, m_j, \text{Task}) \)

The prospective collaborator(s) must possess the resources that are required by the task. Negotiations regarding the task execution occur before they enter into collaboration. That is, if \( m_i \) has a task and he needs some resources which are available with \( m_j \), and the negotiations regarding the task execution and mutual benefits were successful, then \( m_i \) and \( m_j \) become collaborators for Task.

6. Member collaboration exit rule:

\( \text{collab_list}[m_i] = \text{collab_list}[m_i] - m_j \)

If the execution of the collaborated task was completed, or was aborted, then the members are free from the bond of collaboration.

4.2 Formalizing a SHG

A SHG can be formalized as SHG = (SHGID, Resources, Tasks, COLLABS). ‘SHGID’ is the unique
identifier for identifying SHGs, ‘Resources’ are the skills, members, machinery, etc, ‘Tasks’ are the business functions that a SHG performs, ‘COLLABS’ represents the collaborations that a SHG has with other SHGs.

Rule 1: SHG task execution rule:

\[ \text{if } \text{SHG}(\text{task, feasible}) \text{ then execute(SHG, task) else find(SHG, collaborators, task)} \]

After a SHG receives a task, it will check if the task can be executed with its existing resources. If the task is feasible, then it executes the task. Else if the task is infeasible, the SHG will join collaborators, and with their help it will execute the task.

Rule 2: SHG collaborations forming rule:

\[ (\text{SHG}_i.\text{Task.needed_resources} = \text{SHG}_i.\text{available_resources}) \land (\text{negotiations(}\text{SHG}_i, \text{SHG}_j) = \text{success}) \Rightarrow \text{collab(SHG}_i, \text{SHG}_j, \text{Task}) \]

The prospective collaborators must possess the resources that are required by the task. Negotiations regarding the task execution occur before they enter into a collaboration. That is, if \(\text{SHG}_i\) has a task and it needs some resources which are available with \(\text{SHG}_j\), and the negotiations regarding the task execution and mutual benefits were successful, then \(\text{SHG}_i\) and \(\text{SHG}_j\) become collaborators with respect to that Task.

Rule 3: SHG collaboration exit rule:

\[ \text{collab_list}[\text{SHG}_i] = \text{collab_list}[\text{SHG}_i] - \text{SHG}_j \]

If the execution of the collaborated task was completed, or was aborted, then the two SHGs are free from the bond of collaboration. For example: if \(\text{SHG}_j\) is a collaborator of \(\text{SHG}_i\), and \(\text{SHG}_j\) is withdrawing from collaboration, Then

\[ \text{collab_list}[\text{SHG}_i] = \text{collab_list}[\text{SHG}_i] - \text{SHG}_j \]

4.3 Formalizing a Task

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_Infra_structure, 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 \(d_1, d_2, d_3, ..., d_n\) 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:

\[ D = \sum_{i=1}^{n} d_i \]

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

\[ T = (l_i, d_i, t_m, n) \]

When \(\Sigma d_m = D\), the task execution is finished.

4.3.1 Task states

If \(\text{complexity}_i\) is the complexity involved in executing the task \(t_i\), that is, \(\text{complexity}_i\) is the complexity
to move the task a subdistance of $d_i$, then total complexity of the task, $\text{Complexity}_T$ is the summation of complexities of all subtasks. If there are $n$ total subtasks, then

$$\text{Complexity}_T = \sum_{i=1}^{n} (t_i \cdot \text{complexity})$$

**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,

$$\text{Collaborated Distance, } CD = \sum_{i=1}^{k} d_i,$$

And, the complexity of collaborated subtasks is :

$$\text{complexity}_{CD} = \sum_{i=1}^{k} (t_i \cdot \text{complexity})$$

**Task-Collaboration:** 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 5: Task collaboration graph](image)

**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, 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.

![Fig 6: Task classification](image)

From the above figure, we can classify tasks as :

1) Independent task, collaborative task, assignable task
2) Partial collaborative task, complete collaborative task
3) Synchronous partial collaborative task, synchronous complete collaborative task, asynchronous partial collaborative task, asynchronous complete collaboration task
4) Co-located synchronous partial collaborative task, co-located synchronous complete collaborative task, co-located asynchronous partial collaborative task, co-located asynchronous complete collaborative task, remote
synchronous partial collaborative task, remote synchronous complete collaborative task, remote asynchronous partial collaborative task, remote asynchronous complete collaborative task.

Every task has an attribute called ‘IsPartitionable’ which specifies whether the task can be partitioned and executed as subtasks at different places, or it cannot be split and has to be executed entirely at one place. This attribute affects the type of collaboration a SHG would choose. If the task is partitionable, then it splits the task into subtasks and assigns to different collaborators. But if the task is not partitionable, then it looks for a single collaborator who can execute the whole task.

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, SHG_k, SHG_l, SHG_m, ...SHG_n, T).

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 collaborator. 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 λ | λ ∈ (0,1).

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

$$S = \bigcup_{i=1}^{n} \{SHG_i \}$$

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, SHG_k, SHG_l, SHG_m, ...SHG_n, T).

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 collaborator. 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 λ | λ ∈ (0,1).

$$\text{SatDegree}(SHG\_ID, Task\_Id) = \lambda \ast W$$

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.
Table 1: Certainty Value $W$ Computation Table

<table>
<thead>
<tr>
<th>Certainty Value $W$</th>
<th>Computation Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commitment</td>
<td>4</td>
</tr>
<tr>
<td>Open-mindedness</td>
<td>3</td>
</tr>
<tr>
<td>Timeliness</td>
<td>5</td>
</tr>
<tr>
<td>Willingness</td>
<td>6</td>
</tr>
<tr>
<td>Perceptive</td>
<td>3</td>
</tr>
<tr>
<td>Skillfulness</td>
<td>4</td>
</tr>
<tr>
<td>Thinking</td>
<td>5</td>
</tr>
<tr>
<td>Comfortability</td>
<td>4</td>
</tr>
<tr>
<td>Adaptiveness</td>
<td>5</td>
</tr>
<tr>
<td>Significance</td>
<td>4</td>
</tr>
<tr>
<td>Experience</td>
<td>3</td>
</tr>
<tr>
<td>Urgency</td>
<td>5</td>
</tr>
<tr>
<td>Optimism</td>
<td>4</td>
</tr>
<tr>
<td>Prognostics</td>
<td>5</td>
</tr>
<tr>
<td>Collaboration</td>
<td>3</td>
</tr>
<tr>
<td>Organization</td>
<td>5</td>
</tr>
<tr>
<td>Efficiency</td>
<td>3</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
</tr>
</tbody>
</table>

Fig. 8: Collaboration Value

4.5 COMPUTATION OF COLLABORATION VALUE [CV]

$CV = \text{heterogeneity} + \text{complexity} + \text{balance} - \text{embeddedness} - \text{dominance} - \text{resource overlap} + \text{network effects}$

The Collaboration Value [CV] is computed from several attributes with each one of them affecting the collaboration value either positively or negatively. Some attributes affect always positively and some do negatively while some others affect depending on other external conditions.

There is a strong relationship between structural aspects of the SHGs and the effectiveness of collaborations.

One main structural attribute is heterogeneity, which is also described as the diversity / compatibility in resource profiles and social aspects of the collaborating SHGs.

There are two kinds of heterogeneity: Type I heterogeneity is about differences in resources and capabilities; and Type II heterogeneity is about social dimensions such as vision, management approach, and governance system. Notably, heterogeneity (either Type I or Type II) augments the interdependency between the SHGs which thus improves the effectiveness of the collaboration.

$$if \text{ overlap}(SHG_i.(\text{res, cap}) , SHG_j.(\text{res, cap}) ) \cup \text{overlap}(SHG_i.(\text{social aspects}) , SHG_j.(\text{social aspects}) ) \equiv \text{min} \Rightarrow \text{heterogeneity}(SHG_i, SHG_j) = \max$$

Another attribute is complexity of collaboration. Usually business collaborations are task oriented and goal driven relationships. Collaborations usually begin on a small set of tasks wherein the collaborating SHGs evaluate each other’s capabilities, and then start collaborations on larger/complex set of tasks. So, as the complexity of the task(s) increases, the need for collaboration also increases. Complexity is normally characterized along two dimensions: the scope and the intensity of collaborations.

Scope is described as the range of services / products that must be produced or managed through the collaboration, and intensity is the degree of direct involvement on the part of the SHGs. For instance, two SHGs collaborating to manufacture a new machine require a higher level of commitment in terms of resources and manpower than a single sale-campaign. An increase in the complexity of the collaboration needs greater coordination mechanisms, resource expenditures, and constant contact between the SHGs on a regular basis.

Complexity of Collaboration, $Collab_Comp = \text{scope of collaboration} + \text{intensity of collaboration}$

Which is: task complexity + degree of involvement

Balance in the collaboration is the collaborators’ view about the fairness of mechanisms, investments, etc in addition to sharing risks and benefits. The relative view of the collaborators about the costs to benefits ratio is the key variable. One collaborator’s view of balance consists of the collaborator’s estimation of equity in resources, the time, and cognitive resources allocated to the partnership. The better the balance, the stronger the collaboration.

$$collaboration\_balance(SHG_i, SHG_j) \Rightarrow \text{count}(SHG_i.\text{tasks}) \equiv \text{count}(SHG_j.\text{tasks}) \land \text{cost}(SHG_i.\text{tasks}) \equiv \text{cost}(SHG_j.\text{tasks}) \land \text{benefit}(SHG_i.\text{tasks}) \equiv \text{benefit}(SHG_j.\text{tasks})$$

If the SHG has collaborated with any other SHG then it would have the collaboration rating. A good collaboration rating will augment the collaboration value. A negative collaboration rating will adversely affect the...
collaboration value. Collaboration rating was already computed above.

Dominance is the degree to which resources (both physical and informational) flow to and from a SHG, and thus reflects the power of the SHG compared to other SHGs in the network. It also acquires the popularity and visibility within the network. Dominance allows SHGs to obtain control over resources, and participate in collaborations. Dominance also helps in improving the extent to which a SHG is able to pursue its own goals. A dominant SHG attracts more preferential linkages and increases subservience of the collaborators. Likewise, a dominant collaborator enhances the stability of the collaboration. The extent to which a SHG is dominant in a network or collaborates with a dominant collaborator does have a positive impact on the collaboration value.

\[ \text{Dominant(SHGi)} \Rightarrow (\text{resFlow(SHGi, Network)} = \max A \text{resFlow(Network, SHGi)} \land \max A \text{infoFlow(SHGi, Network)} \equiv \max A \text{infoFlow(Network, SHGi)} \equiv \max) \]

Embeddedness is the degree to which a SHG is embedded within the network, and the degree of positive experiences a SHG has by taking part in a series of collaborations. Experience in collaborations increases SHGs’ chances of surviving and maintaining existing collaborations. It also provides new capabilities and administrative skills that facilitate managing new collaborations. Further, their position in the network increases their attractiveness to other SHGs and gives a good number of varied contacts. Conversely, the negative argument of embeddedness is that it makes SHGs open to shocks from entities outside of the SHG network or insulating them from resources that exist beyond the network. Hence, networks built on repeated ties can reduce the flexibility, lock in resources, and ultimately lead to inferior quality.

\[ \text{Emb(SHGi)} = \text{collaboration_rating(SHGi)} \cup \text{count(SHGi, collaborations)} \cup \text{risk(SHGi, openness_to_external_shocks)} \]

Degree of resource overlaps defines the extents to which other SHGs have similar resources. Overlap has considerable impact on the competitive intensity in the business market and confines individual SHGs in their economic action. Even though resource overlap actually increases competitive intensity and thus compels SHGs’ for improvement, it means a negative impact on the collaboration.

\[ \text{Res_overlap} = \text{overlap(SHGi.res, SHGnetwork.res)} \]

Reachability_overhead is the cost involved in reaching the prospective collaborator. Sometimes it could be too large that one SHG would have found a perfect collaborator, but has to discard the collaboration because of the immensity of the reachability overhead.

### 4.6 ARCHITECTURE FOR COLLABORATIVE PRODUCT DEVELOPMENT IN SHGS

SHG’s architecture for collaborative product development [CPD] consists of collaborative product design, simulation and testing and evaluation and internet-based collaborative platforms.

Requirements for Collaborative Product Development in SHGs: The collaborative product development requires skills from product designers and trade experts in multiple disciplines and includes entire lifecycle from market (custom needs and product supply) to market (sale, use, maintenance, and disposal) and every stage being an iterative process and includes some sub-processes, tasks, and activities. For example, during product specification, collaborative design, collaborative simulation and product testing are involved in the product design stage. In order to improve effectiveness and reduce design time, not only three sub-process should collaborate and support each other, but also all stages of lifecycle should support each other.

Some basic requirements of collaborative product development framework are system architecture, product design, process modeling and management, communication and collaboration, and implementation methodologies and tools.

![Fig. 9: Design, Simulation and testing and Evaluation Relationship](image)

**System architecture:** The multidisciplinary nature of collaborative product development requires various activities. According to Roy et al. three principal modes of collaboration in a collaborative product development framework are: (1) Human-to-Human collaboration (2) Design Service-to-Design Service collaboration (3) Human-to-Design Service collaboration.
Product design: Product design lies at the heart of collaborative product development. The design solution should not only support display of product prototypes but also geometric editing of the prototypes by other experts. A preferable approach would be one that enables the modification of product prototypes directly, and displays the modification procedures to all the cooperating product designers. The solution should also support and facilitate the integration between prototype designing, analysis and simulation, test and evaluation.

Process modeling and management: Collaborative product development process involves distributed information and hierarchically nested alternatives. Managing collaborative product development process is not simply solving a set of equations or finding an optimal combination of parameters. Effective process modeling and management is required for exchanging and sharing of collaborative product development information between participants.

Communication & collaboration: Communication is very essential for effective concurrent engineering. In product development, several things keep changing - perhaps a design requirement, an unanticipated simulation or test result, the availability of a component, or an improvement to the manufacturing process. Qualitative product development and productivity requires quick communication about such changes and getting the information to the right place. Designers need to assess their ideas and decisions, and to notify the affected parts in an appropriate way; so both synchronous and asynchronous communications are needed.

Implementation methodologies and tools: An important area for improvement concerns validation of theories and tools. Empirical studies with a thorough methodological basis are suited for this purpose. So research is needed on the impact of concurrent engineering solutions on the performance of real businesses. Suitable collaborative product development methodologies are helpful and critical for better understanding and successful implementation.

This paper considered the above five requirements because engineering design is of great significance to collaborative product development process. There are still some other requirements of collaborative product development, e.g., security, knowledge management, and intelligent interface. It can be observed that these five requirements relate, support and facilitate each other. That is one of reasons why implementation of collaborative product development is so complex in practice.

For SHGs, mature commercial software tools and integration technologies are preferable choices. Some academic prototype systems are not suitable for them to choose. Clearly, because of the system complexity and particularity, no off-the-shelf solutions are available for SHGs.

5. CONCLUSIONS AND PERSPECTIVES

This paper formalizes a member belonging to a SHG. The attributes of a member, and how the member forms collaborations, exits from collaborations are also detailed. It formalizes a SHG and specifies its collaboration rules. 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 value is also described. This paper also discusses the requirements of collaborative product development in SHGs and proposes a solution using an Internet-based collaborative platform. In the middle layer of proposed architecture, process management module, product structure management module, VSM module, and multimedia conference module work together for product development team members (primary users) as well as production, marketing, staff and company managers, suppliers and customers (secondary users). Some commercial tools are integrated in the proposed architecture. New technologies make the architecture open and extensible. In future, research has to focused on the implementation of proposed modules and a prototype system. And, integration of whole collaborative product development process in SHGs is our final goal. In summary, the proposed collaborative product development solution focuses on reusing and reorganizing the existing resources in SHGs to a greater extent and minimizing the change in the working environment thus reducing investment on system development.

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


[15] Multi-Agent Cooperation Based on Interest Group, Zhigang Huang, Pengyi Gao, Yan He, Qing He, International Conference on Computational Intelligence and Software Engineering (CiSE ’09), 2009.

[16] Cooperation Protocols in Multi-Agent Robotic Systems, FANG- CHANG LIN AND JANE YUNG - JEN HSU


