CHAPTER 4

BEHAVIOR MODELING IN DISTRIBUTED ENVIRONMENT

This chapter focuses on a Behavior Model, which attempts to categorize behavioral aspects of each collaborator taking part in a group activity, in order to achieve good collaboration and cooperation in solving problem's with common global goals. Good coordination among members of a group can be achieved by acquisition and use of knowledge about oneself and about other collaborators of the group. Knowledge about oneself and other members working in collaboration is accumulated, analyzed, and quantified in the form of trust factors. In this work, cooperative, noncooperative, and partially cooperative problem solving environment have been monitored by suitably altering the Behavior Model to account for the changing problem solving environment.

4.1 INTRODUCTION

Most of the approaches in DAI (Avouris 1992, Uma 1996a, Genesereth 1986), proceeds with the assumption that the domain is cooperative, and deals with various other aspects like reasoning, representation, resolving disparity, etc. In cooperative environment, the cooperation and coordination issues become simple, for the reason that, the co-collaborator is assumed to be reliable and the information given by them is used as it is. This work attempts to solve cooperation and coordination especially under noncooperative environment. In this work, even when the domain is cooperative, effectiveness of the coordination is enhanced by
associating a trust factor given by the collaborator itself with the information (partial results or subgoals) presented to the group by it. In cooperative environment, the trust factor is calculated locally by each collaborator about its own data. This trust factor is called as Local Trust Factor (LTF) (Anandakumar 1997a). This essentially means that, each collaborator reveals the quality of its own contribution, which can be taken into consideration by the group as a whole, in arriving at an acceptable solution. But such is not the case when domain changes from cooperative to noncooperative. The assumption that the trust factors supplied by the co-collaborators can be used as such is removed when the domain changes to noncooperative. Therefore, it now becomes the responsibility of each collaborator to keep track of capabilities, qualities, and contribution of each co-collaborator with whom it would be interacting. In order to tackle this issue, the Behavior Model in this work acquires, analyses, and quantifies knowledge about other co-collaborators in the form of Behavior Trust Factor (associated with each collaborator is a corresponding BTF) (Anandakumar 1998b). It is this trust factor that is used by every collaborator to evaluate the information (partial results/subgoals) provided by each co-collaborator. In the case of partially cooperative environment, each collaborator uses the LTF sent by each co-collaborator and also the BTF associated with each collaborator, in order to judge the information.

The Behavior Model essentially uses four functional local agents. The Data Analysis Agent (DAA) and Trust Computing Agent (TCA) are used for computing of LTF. The DAA analyses the local databases and passes the summary reports to TCA, which then computes LTF. The TCA uses statistical methods, which helps in reasoning the variations in LTFs among the collaborators. The Behavior Model uses the Similarity Analysis Agent (SAA) to analyze knowledge acquired (received) about (from) the co-collaborators.
and the Behavior Analysis Agent (BAA) calculates the BTF associated with each collaborator. All these agents are explained in detail in the following sections.

4.2 LOCAL TRUST FACTOR

In collaborative work, the degree of confidence group members have about each other plays an important role in many issues like decision making, understanding, resolving disparity, etc. In order to make good decision or judgement, each collaborator should have information about contents, organization, and distribution about its own data. Such self-diagnosis is a very essential characteristic for good collaboration among groups of human beings (Sathprakashanada 1998). In this work, every data or result or decision taken by each collaborator is analyzed and a self-confidence factor or trust factor LTF is associated with the above information. This helps other collaborators to judge each other's information, data, results, etc. The collaborator recomputes this LTF after every problem is solved. In this work, the self-diagnosis is carried out by the DAA, and LTF by TCA, which are explained below.

4.2.1 Data Analysis Agent (DAA)

Most of the systems store information in their databases (past and present data). That is, by analyzing the databases, the system can know more about itself like situation tackled, activities etc., and about the other systems (who are worked with it). Thus, data analysis plays an important role in gaining more knowledge about the particular systems (here collaborators) who are worked with it. Hence, there is growing realization and expectation that data, intelligently analyzed and presented, will be a valuable resource to be used for competitive advantage. The computer science community is responding to both the scientific and practical challenges presented by the
need to find the knowledge drift in the flood of data (Frawley 1991). In this thesis, it is assumed that, each collaborator has access to three databases: project database, skill base, and knowledge base (the nature of data in each of them is application dependent). Data analysis is carried out on local knowledge available in these databases present in the collaborator's repository in order to acquire meta-knowledge about itself, and this is used to calculate the amount of belief (trust) it can be put on its own data. Each record of the relational database is an ordered list of values, one value for each field. In this work, each record value is known as pattern. For example, in global software development application considered in this work, project-database contains project details: work order no, project_name, starting date, ending date, project type, the module details etc. Similarly, details about the other databases are assumed to be available. A method for LTF calculation is suggested (Anandakumar 1997a), which is based on data analysis of local databases. The DAA applies scanning process to the entire database provided time is not a constraint, or to a portion of the database where time is a constraint factor.

Initially the entire database is scanned assuming time is not a constraint. Let $A_1, A_2, ..., A_K$ be the total number ($K$) of attributes available in project database, and $V_1, V_2, ..., V_K$ be the set of corresponding values of the attribute. The DAA scans the database vertically, and produces the summary reports. The summary reports of each attribute $A_k$ has the following information, the $D$ distinct values (or patterns) and their corresponding frequency $F_{P_{dk}}$, the frequency of occurrence of $d^{th}$ pattern of $k^{th}$ attribute. These reports are stored for later usage. The heuristic values for the $d^{th}$ pattern $P_{d_{Ak}}$ of $k^{th}$ attribute $A_k$ is computed using the following equation assuming the patterns are uniformly distributed, which implies that all patterns occur with equal probabilities. $HV_{d_{Ak}}$ is the heuristic value which acts
as upper limit of each $d^{th}$ pattern $P_{dk}$ of $k^{th}$ attribute which is calculated as follows:

$$HV_{dk} = \frac{\sum_{d=1}^{D} FP_{dk}}{D}$$

Here it is assumed that, all attributes are equally important. Thus, it reflects the overall confidence that the collaborator has on its own data, available in the databases. If the collaborator wants to know about some specific part of the database or some sample size (if time is constraint) it can use the same procedure, which is applied with this new constraint. The DAA uses statistical inferences or relationships to set sample size $ss$ (to be scanned), for other collaborators, when they requested for information like LTF of specific information, partial result etc.

While scanning the database, the DAA also produces a number of other outputs like, Auto-Biographical Profile (ABP) of its own and the Biographical Profiles (BP) for known co-collaborators with available information. The ABP contains Name, Experience, LTF value, No-of-Projects handled, No-of-individual projects, Area_of_interest, Client_list, Project Leader, type of projects etc. For example, the Area_of_interest is a domain with the attributes <Fields, SW_type, Area>. Some attribute values are given below:

- Field $\rightarrow$ Animation / Radar / Networking / Compiler_design etc.
- SW_type $\rightarrow$ System_software / Application_software / Networking & Communication etc.
- Area $\rightarrow$ Engg. / Industrial / Defense / Networking etc.
In this work, same structure is maintained for both the profiles (Auto-Biographical and Biographical). The DAA supply the above outputs to other local functional agents when they request for it. The same procedure is also applicable to the other two databases.

4.2.2 Trust Computing Agent (TCA)

Trust Computing Agent (TCA) receive frequency reports from DAA, for further process. Here, a statistical model is provided for self-confidence or trust factor. This self-confidence or trust factor is measured by Local Trust Factor (LTF), and is calculated by every collaborator by analyzing its own databases. The TCA receives summary reports from DAA. From these reports, the Trust Factor $TF_{dk}$ is computed based on pattern $P_{dk}$ for $d^{th}$ pattern $P_{dk}$ of $k^{th}$ attribute $A_k$ assuming that, if the pattern has not occurred at all i.e. $FP_{dk} = 0$ then the trust factor $TF_{dk}$ is zero. This is because nothing can be said about the local trust worthiness of a pattern not yet encountered. The upper limit of the $TF_{dk}$ for a pattern $P_{dk}$ is fixed to $HV_{dk}$ (calculated by DAA) due to the assumption of uniform distribution. Full trust is assumed if the upper limit is exceeded. The $TF_{dk}$ is computed using the equation:

$$TF_{dk} = \begin{cases} 
0 & \text{if } FP_{dk} = 0 \\
FP_{dk}/HV_{dk} & \text{if } 0 < FP_{dk} \leq HV_{dk} \\
1.0 & \text{if } FP_{dk} > HV_{dk} 
\end{cases}$$

(4.2)

Here $FP_{dk}$ is the frequency of occurrence of the pattern $P_{dk}$ of $k^{th}$ attribute $A_k$. The statistical model to compute LTF is given below.

$$LTF = \frac{1}{K} \sum_{k=1}^{K} \left( \frac{1}{D} \sum_{d=1}^{D} TF_{dk} \right)$$

(4.3)

Every collaborator calculates the LTF periodically (here after every problem)
to account dynamically changing information. The TCA can supply the LTF of entire database or LTF of specific information depending on the sample size set by the DAA. Thus, LTF reflects the confidence of the entire database or particular part of the database. LTF samples are given below.

**Table 4.1 LTFs of C₁ recorded at time t**

<table>
<thead>
<tr>
<th>Time</th>
<th>13</th>
<th>28</th>
<th>42</th>
<th>55</th>
<th>68</th>
<th>78</th>
<th>106</th>
<th>120</th>
<th>135</th>
<th>150</th>
<th>164</th>
<th>181</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTF</td>
<td>0.616</td>
<td>0.616</td>
<td>0.616</td>
<td>0.545</td>
<td>0.546</td>
<td>0.538</td>
<td>0.538</td>
<td>0.538</td>
<td>0.538</td>
<td>0.538</td>
<td>0.538</td>
<td>0.538</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0.625</th>
<th>0.746</th>
<th>0.746</th>
<th>0.76</th>
<th>0.746</th>
<th>0.746</th>
<th>0.746</th>
<th>0.841</th>
</tr>
</thead>
<tbody>
<tr>
<td>141</td>
<td>207</td>
<td>224</td>
<td>240</td>
<td>256</td>
<td>272</td>
<td>288</td>
<td>369</td>
</tr>
</tbody>
</table>

**Table 4.2 LTFs of the collaborator C₃**

<table>
<thead>
<tr>
<th>Time</th>
<th>6</th>
<th>11</th>
<th>19</th>
<th>28</th>
<th>35</th>
<th>50</th>
<th>63</th>
<th>71</th>
<th>80</th>
<th>85</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTF</td>
<td>0.412</td>
<td>0.412</td>
<td>0.342</td>
<td>0.379</td>
<td>0.400</td>
<td>0.400</td>
<td>0.417</td>
<td>0.250</td>
<td>0.250</td>
<td>0.400</td>
<td>0.400</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0.400</th>
<th>0.841</th>
<th>0.841</th>
<th>0.841</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>113</td>
<td>133</td>
<td>149</td>
</tr>
</tbody>
</table>

**Table 4.3 LTFs of the collaborator C₅**

<table>
<thead>
<tr>
<th>Time</th>
<th>21</th>
<th>39</th>
<th>49</th>
<th>59</th>
<th>85</th>
<th>112</th>
<th>137</th>
<th>158</th>
<th>227</th>
<th>278</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTF</td>
<td>0.412</td>
<td>0.412</td>
<td>0.558</td>
<td>0.558</td>
<td>0.558</td>
<td>0.648</td>
<td>0.846</td>
<td>0.846</td>
<td>0.846</td>
<td>0.846</td>
</tr>
</tbody>
</table>

The local collaborators C₁'s LTFs are recorded at Time T is shown in table 4.1. Table 4.2 and 4.3 shows the received LTFs of C₃ and C₅. The plots for the above collaborators are drawn for analysis purpose [Figures 4.1, 4.2 and 4.3].
Figure 4.1. Self confidence of the local collaborator $C_t$

Figure 4.2. Self confidence of the collaborator $C_3$

Figure 4.3. Self confidence of the collaborator $C_s$
This LTF is broadcasted to other collaborators periodically or when the request is received. In this work, each collaborator attaches this LTF to information before sending. The broadcasted LTF is used as a basis on which collaborators can arrive at some agreement among themselves. The collaborator having highest LTF is automatically selected as the leader of the group (if necessary). This LTF also helps in forming hierarchical groups automatically. For example (see figure 4.4), let CC is collaborator set and corresponding LTFs are stored in set L. That is, CC = {C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9}, L = {0.8, 0.4, 0.6, 0.9, 0.6, 0.7, 0.8, 0.65, 0.8}. Now, the hierarchical group is formed automatically. The collaborator C_4 belongs to level-0, collaborators C_1, C_7, and C_9 forms the subset in level-1, and C_6 in level-2, C_8 in level-3, C_3 and C_5 are in level-4, and finally C_2 in level-5. If the number of levels reduced or restricted, then the lower level subgroup is combined with the previous level. Here, the group leader is assumed. For example, in level 1, all the collaborators has the same LTF values, from the Fig. 4.4, C7 is

![Figure 4.4 Hierarchical groups](image-url)
selected as a group leader for the next level. Otherwise, voting, (Garcia - Molina 1988, Tanenbaum 1995) method may be used to select the leader. The TCA uses statistical methods (Kaplan, 1987, Hogg 1977) to compare other collaborators' LTFs with its own LTF. The statistical inference is described in the following section.

4.2.2.1 Statistical Inference

T-test technique is used to select between null hypothesis H0 or alternative hypothesis H1, which is then used for comparison of a pair of collaborators. H0 / H1 can be framed appropriately based on the current situation and is usually determined by mean (μ) or variance (S) comparison. For example, H0: Both the Collaborator's LTFs are same, and H1: Not same. The t-test (tv) is used to know whether two sets are significantly different or not. The t-test (tv) is computed as follows:

\[
tv = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{(N_1 - 1) S_1^2 + (N_2 - 1) S_2^2}{N_1 + N_2 - 2} \left[ \frac{1}{N_1} + \frac{1}{N_2} \right]}}
\]  

(4.4)

where \(X_1, X_2\) are the sets of LTFs. Here, \(X_1\) in the set of LTFs of the local collaborator and \(X_2\) is other collaborator's (say \(C_3\)) LTFs. \(N_1\) and \(N_2\) are the number of terms present, and \(\bar{X}_1\) and \(\bar{X}_2\) are the means of the corresponding sets. The variance \(S_s\) of a set \(X_s\) is

\[
S_s = \frac{\sum X_i^2 - (\sum X_i)^2}{N_s - 1}
\]

(4.5)

Here \(s = 1\) and 2. The decision rule is stated as follows:
TRule3: if tv value < - sl or tv value > + sl then reject H0 and accept H1.

where sl is the required significance level.

The term correlation (r) indicates the relationships between the two variables \( X_1 \) and \( X_2 \) in which, changes in the values of one variable, the values of the other variable also changes. This reduces the range of uncertainty associated with decision making. Coefficient of correlation is calculated to study the extent or degree of correlation between two variables. The coefficient of correlation \( r \) of the two variables \( X_1 \) and \( X_2 \) is given by the formula:

\[
r = \frac{N_i \sum X_1 X_2 - (\sum X_1)(\sum X_2)}{\sqrt{[N_i \sum X_1^2 - (\sum X_1)^2][N_i \sum X_2^2 - (\sum X_2)^2]}} \tag{4.6}
\]

where \( N_i \) is the number of terms in \( X_i \). The standard deviation is calculated for each variable \( X_i \) corresponding to the set of each collaborator and is calculated as

\[
\sigma_{x_i} = \sqrt{\frac{\sum X_i^2 - (\sum X_i)^2}{N_i}} \tag{4.7}
\]

where \( N_i \) corresponds to number of terms in each \( X_i \). The coefficient of correlation measures the degree of association, and standard deviation gives the measure of dispersion. From Table 4.1, 4.2 and 4.3, the correlation's are found. The values are shown below for the Exponential curve fit.

- Exponential fit: \( y = ae^{bx} \)
- Standard error: 0.124686
- Correlation \( r \): 0.4128
and coefficients are \( a = 0.62849 \) and \( b = 0.0079 \).

From Table 4.2

- Standard error = 0.1399,
- Correlation (r) = 0.739
- Coefficient \( a \) = 0.243 and \( b \) = 0.0082

From Table 4.3

- Standard error = 0.0676
- Correlation (r) = 0.7329
- Coefficient \( a \) = 0.4856 and \( b \) = 0.00132

Now, the local collaborator \( C_i \) predicts the future LTF values for other collaborators.

### 4.3 BEHAVIOR MODELING

The models may be derived from first principles, identified from data, or complied via knowledge engineering (Hudlicka 1987, Ramussen 1983, Lesser 1980). Some of the potential benefits of modeling other collaborators are the ability: (i) to predict the behavior of other collaborators, (ii) to help in its own internal planning and coordination, (iii) to reason about other collaborators, and (iv) to enable coordination without communication (Genesereth 1986). A model may be a very simple, being used to reason about only one aspect of another collaborator's behavior, or it may be extremely complex, allowing the collaborator to reason about how to influence another collaborator by some communication act or other means. Rouse and Hammer (1991) discussed the modeling limits on intelligent systems. They provided a general framework for considering the role in intelligent system of model...
of physical, behavioral and operational phenomena. They identified ten types of modeling limits and discussed their consequences. Very little work has done in modeling intelligent system behavior. Since, in this work, experience and accumulated knowledge plays an important role, a statistical model has been used to account for other collaborator's behavior. The collaborator's behavior is represented by a trust factor BTF, which is computed by the Behavior Analysis Agent (BAA). Similarity analysis helps to analyze behavior like accuracy of the partial results, skill used in problem solving, reliability values etc. In this work, a local functional agent called Similarity Analysis Agent is introduced, which performs the necessary similarity analysis. The following sections describe the SAA and BAA.

4.3.1 Similarity Analysis Agent (SAA)

The similarity analysis (Anandakumar 1997a) is performed to acquire and process information about the collaborators. It plays a fundamental role in theories of knowledge and behavior. It serve as an organizing principle by which individuals classify objects, form concepts, and make generalization indeed, the concept of similarity is ubiquitous in psychological theory. Similarity or dissimilarity of data appears in different forms: ratings of pairs, sorting of objects, communality between associations, errors of substitution, and correlation between occurrences. The SAA performs similarity study to acquire information about the collaborator itself and about other co-collaborators. The SAA applies the similarity process to profiles and to newly generated subgoals or partial results. The profile matching is applied to project requirement profile, Auto-Biographical profiles, and Biographical profiles. To simplify the process, a completely similar structure for subgoals and partial results with the same parameters is assumed. The similarity process compares the features (properties or characteristics) of the matching (Tuersly 1977) objects. The feature matching principle is explained in the following section.
4.3.1.1 Feature Matching

Let $FM = \{a, b, c, \ldots\}$ be the domain of objects under study. Assume that each object in $FM$ is represented by a set of features or attributes, and let $A, B, C,$ denote the sets of features associated with the objects $a, b, c$ respectively. For example, in this work, the features of profile (Project Requirement profile / Auto-Biographical profile/ Biographical profile) represent concrete properties such as experience, skill, project type, project starting date, etc. Then, let $s(a, b)$ be a measure of the similarity of $a$ to $b$ defined for all distinct $a, b$ in $FM$. The scale $s$ is treated as an ordinal measure of similarity. That is, $s(a, b) > s(c, d)$ means that $a$ is more similar to $b$ than $c$ is to $d$. The present theory is based on the following assumption.

$$s(a, b) = F(A \cap B, A-B, B-A) \quad (4.8)$$

The similarity of $a$ to $b$ is expressed as a function $F$ of three arguments: $A \cap B$, the features that are common to both $a$ and $b$; $A-B$, the features that belong to $a$ but not to $b$; $B-A$, the features that belong to $b$ but not to $a$.

$$s(a, b) \geq s(a, c) \text{ whenever }$$

$$A \cap B \supset A \cap C, A-B \subseteq A-C, \text{ and } B-A \subseteq C-A$$

That is, similarity increases with addition of common features and/or deletion of distinctive features (i.e. features that are belongs to one object but not to other). In order to control the progress of similarity process, the SAA uses heuristics called progressive heuristics.

4.3.1.2 Similarity Study and Progressive Heuristics

The SAA performs the similarity study in two major steps.

i) Comparison and Analysis of local information, and

ii) Comparison and Analysis with other collaborators' information
The later step has less priority than the former, and both the steps are carried out locally. When the collaborator generates a subgoal (partial results) locally or receives a subgoal (partial results) from the other collaborator, the SAA compares the previously encountered goals (partial results) with the current goal (partial results). The percentage matches being available, the mean ($\mu$) and standard deviation ($\sigma$) for the current goal is obtained. The SAA then decides whether the similarity process is sufficient, else the similarity match for the current subgoal with other collaborator's subgoal is performed. Various heuristic values are used which are dynamic in nature and are used to control the process of similarity study. Depending on these heuristics values the SAA can abandon the process after a minimum of effort. As the collaborator gains experience, these heuristic values are also refined. Hence, they are called as progressive heuristics (explained below).

This similarity study in general gives raise to 3 different Cases:

Case A: All have the same / similar subgoal (or partial result)
Case B: None of the collaborator has the subgoal (or partial result)
Case C: At least one but not all collaborator has the subgoal (or partial result).

Having done the similarity study, the SAA records the information such as degree of similarity, accuracy of partial results, etc. This information helps the collaborator to contact appropriate co-collaborators (if more than one collaborator has solved the problem), to set an accuracy level (depending on the degree of similarity and/or accuracy) etc.

4.3.1.2.1 Progressive Heuristics

Heuristics play an important role in problem solving and search and helps in finding the solution faster (i.e. sub optimal solution). The solution so
obtained may however not always be optimal. Here, the SAA uses the following heuristic values called as progressive heuristics to control the similarity matching process.

i) HF1 and HF2 are heuristics, which acts as lower and upper limits for the similarity process

ii) HC is the heuristic counter for repeated activity.

To reduce repetitive operation, a heuristic counter (HC) indicating frequency of goal solving is maintained by the SAA for each sub goal. As soon as the counter variable crosses the predefined value (which may be different for different subgoals), the system neglects the received result as it already exists or sends the previously computed result to the corresponding collaborator. The heuristic rules are formulated using the above mentioned heuristic variables. Associated with a subgoal whose match with another goal is to be determined is an accepted level of match ($sl$) between two subgoals of different collaborators, $sl$ varies with each collaborator pair. This $sl$ along with the mean ($\mu$) and standard deviation ($\sigma$) calculated in the similarity process is used to determine the confidence intervals [HF1, HF2]. HF1 and HF2 acts as lower and upper boundaries for percentage match and are computed using the equations:

\[
\begin{align*}
HF1 & = \mu - sl \left( \frac{\sigma}{n-1} \right) \\
HF2 & = \mu + sl \left( \frac{\sigma}{n-1} \right)
\end{align*}
\]

where $n$ is the number of subgoals involved in similarity match process.

When a collaborator solves the same subgoal frequently, the accuracy of the solution normally increases and the same collaborator can be used to solve that particular subgoal, without going for similarity match process. The
heuristic rules for similarity analysis formulated using the above heuristic variables are given below.

R1: if (degree of similarity is < HF1) then
   "DISSIMILAR OR UNKNOWN SUBGOAL".
R2: if (degree of similarity is > HF2) then
   "SIMILAR SUBGOAL".
R3: if (degree of similarity lies between HF1 and HF2) then
   "PARTIALLY SIMILAR SUBGOAL".
R4: if (counter > HC) then
   "REPEATED SUBGOAL".

4.3.1.3 Profile Matching

In the global software development application considered in this work, the SAA performs the similarity analysis between the profiles to know more about co-collaborator's skills, experience, project type, number of problems solved etc. Here, the SAA uses the similarity method suggested by Hadizikadic (1997) to compare profiles. Let ABP be the collaborator's own Auto-Biographical profile, and BP the Biographical profile of the co-collaborator. Both the profiles are the output of Data Analysis Agent (DAA). To simplify the matching problem, in this work, all features of the profiles are given with equal priority. The SAA find common features, difference of the features, total number of instances etc., of both the profiles. Then, it computes the percentage of similarity of ABP with BP, and vice versa. The SAA uses the following rules to classify the profiles into different categories.

SARule1: IF percentage of similarity ≥ fz1 and above THEN "Exactly Similar".
SARule2: IF fz2 ≤ percentage of similarity < fz1 THEN "Partially Similar".
SARule3: IF percentage of similarity is < fz2 THEN "Dissimilar".
Here, f1 and f2 are the heuristic variables, which are set by SAA depending on the degree of similarity needed. Similarly, the above process is repeated for all partial results. The entire process can be repeated for all co-collaborators. The computed results are listed below.

Cj with C2 Similarity = 0.660041, Dissimilarity = 0.317318 time taken = 20
Cj with C3 Similarity = 0.663263, Dissimilarity = 0.312788 time taken = 21
Cj with C4 Similarity = 0.17872, Dissimilarity = 0.823887 time taken = 12
Cj with C5 Similarity = 0.990041, Dissimilarity = 0.017318 time taken = 19
Cj with C6 Similarity = 0.963263, Dissimilarity = 0.042788 time taken = 19
Cj with C7 Similarity = 0.07872, Dissimilarity = 0.923887 time taken = 10
Cj with C8 Similarity = 0.560041, Dissimilarity = 0.427318 time taken = 20
Cj with C9 Similarity = 0.603263, Dissimilarity = 0.412788 time taken = 19
Cj with C10 Similarity = 0.57872, Dissimilarity = 0.423887 time taken = 13

The appropriate similarity results are passed to Behavior Analysis Agent for BTF computation.

4.4 BEHAVIOR ANALYSIS AGENT (BAA)

In an open and dynamic environment the collaborators behavior play an important role, like prioritizing the information given by the other collaborators, how much trust can be put on the supplied information (subgoal or partial results), knowledge, or services etc. Here, each collaborator is attached with Behavior Trust Factor (BTF): which is used to study the behavior of co-collaborator, in order to improve the group
effectiveness. Initially the BTF of its own is set to 1.0, whereas the co-collaborator's is set to zero. The BTF is then recalculated after every problem is solved to account dynamically changing behavior.

4.4.1 BTF Computation

A number of variables can be used to define the human behavior or characteristics. In this work, the co-collaborator's behavior is defined by considering twelve such parameters. These parameters are called as personality parameters (Robbins, 1996, Robey 1981) and are explained below. Here, a statistical model is described with the help of these personality parameters to compute BTF of other collaborators. The parameters are classified as absolutely necessary, supporting, and other parameters. In this work, all the twelve personality parameters are treated with equal priority, and the statistical model is described by:

\[
\text{BTF} = \frac{1}{6} \left[ \beta_1 \ast \text{LOAD} + \beta_2 \ast \text{INTEREST} + \beta_3 \ast \text{CAPABILITY} + \beta_4 \ast \text{TECHKNOWN} + \beta_5 \ast \text{TIMEMAN} + \beta_6 \ast \text{SITUATION} \right]
\]

\[
\text{CAPABILITY} = \frac{1}{6} \left[ \beta_7 \ast \text{JUDGEMENT} + \beta_8 \ast \text{RESPONSE} + \beta_9 \ast \text{INITIATIVE} + \beta_{10} \ast \text{REASONING} + \beta_{11} \ast \text{CREATIVITY} + \beta_{12} \ast \text{ANALYTICAL} \right]
\]
where \( \beta \), \( 1 \leq x \leq 12 \), is the priority value, and is set depending on the importance of the parameters. The coefficient \( \beta \) is set to 1.0 for \( 1 \leq x \leq 12 \). All these personality parameters are computed locally by the collaborator about the other collaborator’s behavior. The BAA uses the above statistical model to computes BTF for all of its co-collaborator.

In this work, the personality parameters are defined with respect to a global software development application. Here, each collaborator is attached with a project database, skill base and knowledge base. The first parameter LOAD depends on actual load (total number of subtask in hand) and capacity (maximum number of subtask it can handle) of the other collaborators. Both the values are assumed to be known and LOAD is given by

\[
LOAD = \begin{cases} 
1.0 & \text{if (load} \geq \text{capacity)} \\
\frac{load}{capacity} & \text{Otherwise}
\end{cases} 
\]

(4.10)

INTEREST is the second parameter, depends on three factors, they are number of offers accepted by the co-collaborator, percentage of similarity of partial results, and reaction of catalyst agent’s service by it. Then, this parameter is computed by the equation:

\[
INTEREST = \frac{\text{No.} - \text{offers} - \text{accepted}}{\text{No.} - \text{offers} - \text{made}}
\]

(4.11)

Third personality parameter is the TECHKNOWN reflects the technical knowledge known to that particular co-collaborator. It is the ratio of total number of skills to the total number of problems (projects) solved by
it. Therefore

\[ \text{TECHKNOWN} = \frac{\text{No. - skills}}{\text{Total - No. - projects}} \]  \hspace{1cm} (4.12)

Another important personality parameter is TIMEMAN, which indicates the time management capability of the co-collaborator. That is, it checks whether the co-collaborator completes the assigned problem (here project) in time or not. In this work, project time and the catalyst agent's reaction time is considered to define this personality parameter, and is given by

\[ PT = \frac{(0.6*EP + 0.4*IP - 0.1*LP)}{\text{Total - No. - Projects}} \]  \hspace{1cm} (4.13)

where LP is the total number of late projects,
IP is the total number of projects completed in time, and
EP is the total number of early-completed projects.

\[ \text{CAT} = \begin{cases} 0 & \text{if } \text{NCAT} = 0 \\ \frac{(\text{NCAT} - \text{LCAT})}{\text{NCAT}} & \text{Otherwise} \end{cases} \]  \hspace{1cm} (4.14a)

where CAT is the catalyst agents' time,
NCAT is the total number of catalyst agents communicated to Cj, and
LCAT is the total number of late response.

\[ \text{TIMEMAN} = \frac{(PT - CAT)}{2} \]  \hspace{1cm} (4.14b)
Two types of situations are considered in this work. They are Critical and Normal, and are defined by load and capacity of the collaborator of its own. This is to know under what situation the co-collaborator has helped. Thus, the personality parameter SITUATION is defined as

\[
SITUATION = \begin{cases} 
1.0 \text{ (Critical) if load } \geq \text{ capacity} \\
0.5 \text{ (Normal) if load } < \text{ capacity}
\end{cases}
\]  

(4.15)

CAPABILITY is another important and complex personality parameter. In this work, it depends on six more personality parameters, as defined in the statistical model. The parameter JUDGEMENT is used to measure judgement capability of the co-collaborator. It is depends on the similarity analysis of partial results sent by that particular co-collaborator. The equation shows the priorities assigned to similar and partially similar results in this work. Thus,

\[
\text{NPSR} = \text{NR} - (\text{NSR} + \text{NDR})
\]

where NR is the total number of partial results,
NSR is the total number of partial results with percentage of similarity > 0.8,
NDR is the total number of partial results with percentage of similarity < 0.4, and
NPSR is the remaining partial results.

\[
\text{JUDGEMENT} = \left( \frac{(0.6 \ast \text{NSR} + 0.4 \ast \text{NPSR})}{\text{NSR} + \text{NPSR}} \right) \ast \left( \frac{\text{NDR}}{\text{NR}} \right)
\]

(4.16)

The parameter RESPONSE is used to indicate the responsibility of the co-collaborator. It is defined by two variables R1 and R2. The variable R1 is
the ratio of number of times co-collaborator has worked as a leader to total number of projects solved by it. \( R_2 \) gives the co-collaborator's interest as a project leader. Then the personality parameter \( \text{RESPONSE} \) is given by

\[
\text{RESPONSE} = \frac{(R_1 + R_2)}{2} \quad (4.17)
\]

\[
R_1 = \frac{(A/PL)}{\text{Total - No. - Projects}} \quad (4.17a)
\]

\[
PLR = PLO - PLA
\]

\[
R_2 = \begin{cases} 
\frac{(PLA - PLR)}{PLO} & \text{If } (PLO \neq 0) \\
0.0 & \text{Otherwise} 
\end{cases} \quad (4.17b)
\]

where  
- \( A/PL \) is the total number of projects as a project leader,
- \( PLO \) is the total number of offers made for project leader,
- \( PLA \) is the total number of offers accepted as a project leader, and
- \( PLR \) is the total number of offers rejected.

In this work, the co-collaborator's initiative capability is represented by the personality parameter \( \text{INITIATIVE} \), and is measured as follows.

\[
\text{TL} = \frac{\text{INITIATIVE}}{\text{Total-No.-projects}} \quad (4.18)
\]

Where  
- \( TL \) is the total number of projects worked as leader

The co-collaborator's creativity characteristics is defined as ratio of number of new skills used in problem solving to the total number of skills.
known to the co-collaborator. Hence, CREATIVITY is defined as follows.

\[
\text{CREATIVITY} = \frac{\text{No.-of-New skills}}{\text{Total-No.-skills}}
\]  

(4.19)

The analytical ability (ANALYTICAL) of a co-collaborator is defined as a ratio of summation of percentage of similarities of all partial results to the total number of partial results sent by co-collaborator. That is

\[
\text{ANALYTICAL} = \frac{1}{L} \sum_{i=1}^{L} \text{PS}_i
\]  

(4.20)

where PS\(_i\) is the percentage of similarity, and L is the number of partial results sent by co-collaborator

The reasoning capability of the co-collaborator is defined with respect to the penalty made by that particular co-collaborator. In this work, it is measured by the penalty, which is computed depending on the project completion time, and is described as follows:

\[
\text{penalty} = \frac{(\text{float})(\text{PCT} - \text{PT})}{\text{PT}} \quad \text{and} \quad \text{if} (\text{penalty} > 1) \text{ penalty} = 1
\]

\[
\text{PE} = \frac{1}{L} \sum_{i=1}^{L} \text{penalty}_i
\]

\[
\text{PE} = \begin{cases} 
0 & \text{if } (L = 0) \\
\frac{\text{PE}}{T} & \text{Otherwise}
\end{cases}
\]

\[
\text{REASON} = \frac{(T - L)}{T} - \text{PE}
\]  

(4.21)

where  T is the total number of projects
PT is the project duration,
PCT is the project completion time,
L is the number of late projects, and
PE is the total penalty

The BTF is computed after every problem (here project) is completed. The table 4.5 gives the sample output and the corresponding BTF value.

**Table 4.4 Behavior Trust Factors of the Collaborators**

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
<th>$c_5$</th>
<th>$c_6$</th>
<th>$c_7$</th>
<th>$c_8$</th>
<th>$c_9$</th>
<th>$c_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.446</td>
<td>0.302</td>
<td>0.553</td>
<td>0.800</td>
<td>0.800</td>
<td>0.505</td>
<td>0.800</td>
<td>0.800</td>
<td>0.200</td>
<td>0.617</td>
</tr>
<tr>
<td>2</td>
<td>0.705</td>
<td>0.283</td>
<td>0.359</td>
<td>0.800</td>
<td>0.200</td>
<td>0.503</td>
<td>0.591</td>
<td>0.645</td>
<td>0.358</td>
<td>0.200</td>
</tr>
<tr>
<td>3</td>
<td>0.200</td>
<td>0.684</td>
<td>0.566</td>
<td>0.800</td>
<td>0.424</td>
<td>0.352</td>
<td>0.267</td>
<td>0.200</td>
<td>0.622</td>
<td>0.627</td>
</tr>
<tr>
<td>4</td>
<td>0.800</td>
<td>0.293</td>
<td>0.476</td>
<td>0.595</td>
<td>0.352</td>
<td>0.526</td>
<td>0.671</td>
<td>0.302</td>
<td>0.520</td>
<td>0.800</td>
</tr>
<tr>
<td>5</td>
<td>0.552</td>
<td>0.566</td>
<td>0.200</td>
<td>0.541</td>
<td>0.800</td>
<td>0.200</td>
<td>0.552</td>
<td>0.200</td>
<td>0.546</td>
<td>0.375</td>
</tr>
<tr>
<td>6</td>
<td>0.741</td>
<td>0.419</td>
<td>0.616</td>
<td>0.200</td>
<td>0.200</td>
<td>0.520</td>
<td>0.789</td>
<td>0.285</td>
<td>0.200</td>
<td>0.730</td>
</tr>
<tr>
<td>7</td>
<td>0.579</td>
<td>0.307</td>
<td>0.200</td>
<td>0.655</td>
<td>0.263</td>
<td>0.608</td>
<td>0.800</td>
<td>0.800</td>
<td>0.800</td>
<td>0.396</td>
</tr>
<tr>
<td>8</td>
<td>0.638</td>
<td>0.445</td>
<td>0.555</td>
<td>0.691</td>
<td>0.200</td>
<td>0.733</td>
<td>0.800</td>
<td>0.537</td>
<td>0.200</td>
<td>0.800</td>
</tr>
<tr>
<td>9</td>
<td>0.782</td>
<td>0.566</td>
<td>0.292</td>
<td>0.294</td>
<td>0.480</td>
<td>0.557</td>
<td>0.200</td>
<td>0.200</td>
<td>0.200</td>
<td>0.598</td>
</tr>
<tr>
<td>10</td>
<td>0.200</td>
<td>0.488</td>
<td>0.200</td>
<td>0.200</td>
<td>0.200</td>
<td>0.380</td>
<td>0.344</td>
<td>0.800</td>
<td>0.207</td>
<td>0.378</td>
</tr>
</tbody>
</table>

**4.5 ACTUAL TRUST FACTOR (ATF)**

The Behavior Model also uses another trust factor called Actual Trust Factor (ATF) under partially cooperative environment. This trust factor is computed based on LTF and BTF of the particular co-collaborator by giving equal priority. That is, here the collaborator judge the information by considering both the trust factor LTF (sent by co-collaborator) and BTF
(computed locally about co-collaborator). Thus, the ATF of a particular co-collaborator is computed as follows:

\[
AFT = \frac{BTF + LTF}{2}
\]  

(4.22)

4.6 CONCLUSION

In this chapter, a behavior model has been designed for noncooperative environment. The chapter also discuss two statistical models incorporated in the behavior model. They are,

* Statistical model for LTF calculation, and
* Statistical model for BTF calculation.

Agent-based concept used in the model is also highlighted. In the next chapter, a new software agent has been described which helps in the communication mechanism between the collaborators.