

Adjudicator: A Statistical Approach for Learning Ontology Concepts from Peer Agents

Behrouz Far, *Member, IEEE*, Abdel-Halim Hafez Elamy, Nora Houari and Mohsen Afsharchi

Abstract— We present a statistical approach for software agents to learn ontology concepts from peer agents by asking them whether they can reach consensus on significant differences between similar concepts. This method allows agents that are not sharing common ontologies to establish common grounds on concepts known only to some of them, when these common grounds are needed. The method starts with fetching positive and negative examples for a concept vaguely understood by a learner agent from the peer agents. The learner agent then uses a concept learning method to learn the concept in question. Then example objects of the candidate concept are sent back to the peer agents asking for their feedback. Peer agents evaluate the examples using two dimensional rate and weight evaluation criteria. The returned data is then tested for integrity and analyzed using analysis of variance to identify whether a statistical consensus can be achieved among peer agent with respect to the learnt concept. If such consensus exists, the learning agent can add the concept to its ontology with a higher degree of confidence. This will enhance the autonomy and improve communication and cooperation abilities among software agents.

I. INTRODUCTION

HAVING a common syntax, a common semantics and a common context are necessary for communicative agents to interact and understand each other. Ontology research community tries to address issues arisen from violation or relaxation of any of the above three requirements. Conceptualization, that is identifying key concepts and their relations, is central in ontology research. Many researchers have proposed several ways of conceptualization and have devised many forms of conceptual mapping arithmetic [11] [13]. For the sake of simplicity and/or convenience many researchers in their works have assumed that it is possible to establish a common language among agents (e.g., using several variations of agent communication languages, ACL), and also the agents are provided with a complete common understanding of all the concepts they need (e.g., having a common conceptualization). In case that heterogeneity or interoperability is a requirement, many researchers assume

that it is possible to have an already existing common ontology for the agents and that the agent developers can use this common ontology when designing their agents, perhaps by calling an ontology service, thus allowing for easy communication and understanding among the agents. However, the assumption of existence of a common ontology is often too strong or unrealistic. For many application domains, there is no agreement on ontology for the domain among developers. Also for many areas home brewed ontologies already exist and in many cases the potential ontologies are large, unwieldy and encompass more than what a particular agent most probably will ever need and implementing complex ontologies can also easily lead to discrepancies among implementations [1].

A recent approach is to let the agent have their individualized ontologies and provide them with learning and conflict resolution mechanisms for the concepts they need during communication. (See [6] for learning and making up a language; [13] and [1] for learning a concept). The work in [13] has focused on interactions between two agents only and single concepts and [6] was not concerned with concepts. In a previous work Afsharchi and Far have devised a methodology for having agents learn concepts from several peer agents [1]. In this method a learner agent queries peer agents by providing features (and their values) or examples that it thinks are associated with a vaguely understood concept. The queried agents provide the learner with positive and negative examples from their understanding of their own concepts (i.e. known concepts) that seem to fit the query. Then the learner agent uses a learning technique to learn the concept in question [1].

The learner agent must deal with the fact that the peer agents queried might not totally agree on which examples fit the concept and which do not. Therefore, a drawback of the above mentioned method is that there is no guarantee that the peer agents also agree upon the learnt concept. In other words, there still exists some case of misunderstanding due to the learning misses. One solution to this problem is to use a kind of voting mechanism and ask the peer agents vote on the examples for which the contradictory information is obtained. However, voting does not usually resolve the contradictions.

In this paper we have extended the work in [1] by devising a sound statistical method called adjudication that aims at resolving the contradictions among peers by questioning whether they have significantly different viewpoints regarding a concept. In the adjudication method,

This work was supported in part grants from the NSERC and IBM.

Behrouz Far, Dept. of Electrical & Computer Engineering, University of Calgary, Canada, (tel: 403-210-5411; email: far@ucalgary.ca).

Nora Houari, Dept. of Electrical & Computer Engineering, University of Calgary, Canada, (email: nhouari@ucalgary.ca).

Abdel-Halim Hafez Elamy, Dept. of Electrical & Computer Engineering, University of Alberta, Canada, (email:elamy@ualberta.ca).

Mohsen Afsharchi, Dept. of Electrical & Computer Engineering, University of Calgary, Canada, (email: mafsharc@ucalgary.ca).

after a successful learning cycle, example objects of the learnt concept are sent back to the peer agents asking for their feedback. Peer agents will evaluate the examples using a two dimensional rate and weight evaluation criteria. Their response is then tested for integrity and analyzed using analysis of variance (ANOVA) to identify whether a collective statistical consensus can be achieved among peer agent with respect to the learnt concept. If such consensus exists, the learner agent can add the concept to its ontology with a higher degree of confidence.

The structure of this paper is as follows: in Section II we give definitions for the concepts that we use throughout this paper. In Section III the concept learning mechanism is reviewed, Section IV introduces the adjudicator methodology and is followed by an example in Section V. Finally conclusions are drawn in Section VI.

II. TERMINOLOGY AND DEFINITIONS

In this section, we provide definitions for ontologies, concept, dimension, feature, object (example), and agent that we require for our method.

Ontology: We adopt Stumme’s definition (see [12]) who defines *ontology* as a structure $O := (C, \leq_C, R, \sigma, \leq_R)$. Where C and R are two disjoint sets and the members of C are called *concept identifiers* and the members of R are called *relation identifiers*. \leq_C and \leq_R are partial orders on C and R , respectively called *concept hierarchy* or *taxonomy* (\leq_C), and *relation hierarchy* (\leq_R). $\sigma : R \rightarrow C^+$ is a function providing a signature for a relation.

Concept, dimension, feature, object: Many works in databases and machine learning define concepts as collections of objects that share certain feature instantiations. We assume that there exists a set of features $F = \{f_1, f_2, \dots, f_n\}$ and a subset of F is common among agents. This means that the agents have a minimum common ground for communication. We also characterize a concept by using its dimensions and features. Therefore a concept $c \in C$ is represented by a set of dimensions, $c = \{D_1, D_2, \dots, D_m\}$, and each dimension is comprised of certain features, $D_i = \{f_{i1}, f_{i2}, \dots, f_{ik}\}$. Then an object (or example) $o = ([f_1 = v_1], [f_1 = v_2], \dots, [f_n = v_n])$ is characterized by its values for each of the features. For example a concept *software development* can have dimensions such as *modeling*, *process* and *application*. And *process* itself can be comprised of features such as *architecture*, *lifecycle* and *manageability*.

In an ontology, we assign a concept identifier to each symbolic concept that we want to represent in our ontology. The relation \leq_C is supposed to be associated with how concepts are defined. In the literature, taxonomies are often build using the subset relation, i.e. we have $C_i \leq_C C_j$ iff for all objects $o \in C_i$ we have $o \in C_j$. This definition of \leq_C produces a partial order on C and we will use this definition in the following for the ontologies that our agents use.

Agent: in this work we view an agent Ag as a quadruple $Ag = (Sit, Act, Dat, f_{Ag})$. Sit is a set of situations the agent can be in, the representation of a situation naturally depends on the agent’s sensory capabilities. Act is the set of actions

that Ag can perform and Dat is the set of possible values that Ag ’s internal data areas can have. In order to determine its next action, Ag uses the function $f_{Ag} : Sit \times Dat \rightarrow Act$ applied to the current situation and the current values of its internal data areas. The Sit set usually contains parts representing observations of other agents and of the environment the agent is in. In this paper, we assume that among the observations of an agent are all messages send by other agents since the last situation an agent was in.

III. LEARNING NEW CONCEPTS

A goal of this part of research is to develop a method how an agent can learn new concepts for its ontology with the help of other agents. This naturally assumes that the agents do not have the same ontology, otherwise learning would not be necessary. We additionally assume that there are only some base features $F_{base} \subseteq F$ that are known and can be recognized, by all agents and that there are only some base symbolic concepts C_{base} that are known to all agents by name, their feature values for the base features and the objects that are covered by them. Outside of this base common knowledge, individual agents may come with additional features they can recognize and additional concepts they know. Agents might refer to the same such features and concepts by different names and they may have features and concepts that have the same name but are not the same. While all the ontologies used by the agents will use as taxonomy the subset-relation, agents may use different other relations in their ontologies and two agents cannot rely on the same relation identifiers referring to the same relations.

Given this setting, agents will develop problems in working together, since the common ground for communication is too narrow. Our basic idea is to have an agent *learn* required concepts with the *help* of the other agents. Due to the potential differences in the ontologies of agents, objects that are positive and negative examples for a concept will play a major role in teaching an agent a new concept. We assume that the identifying name of an object is a feature in F_{base} . We do not see this as a big limitation, since it is usually not too difficult to establish a clear identification of objects. For example, if the objects are part of the environment, pointing to a particular object is sufficient to identify it.

Although we want all agents to be able to learn new concepts, for explaining our interaction scheme we designate one agent, Ag_L , as the learner agent which wants to learn a new concept and the other agents, Ag_1, \dots, Ag_m , will be its peers. Ag_L has an ontology $O_L = (C_L, \leq_C, R_L, \sigma_L, \leq_R_L)$ and knows a set of features F_L . Analogously, Ag_i has ontology $O_i = (C_i, \leq_C, R_i, \sigma_i, \leq_R_i)$ and knows a set of features F_i . For a concept c known to the agent Ag_i , this agent has in its data areas a set pex_i^c of positive examples for c that it can use to teach c to Ag_L . Part of Act_L are actions QueryConcept, AskClassify, Learn, and Integrate, while part of the Act_i s are the actions FindConcept, CreateNegEx, ReplyQuery, ClassifyEx and ReplyClass; all with

appropriate arguments. These actions form our interaction scheme as:

1. Ag_L determines it needs to know about a concept c_{goal} and performs $QueryConcept("c_{goal}")$ to inform the peer agents about this need.
2. Each agent Ag_i reacts to Ag_L 's query by:
 - (a) performing $FindConcept("c_{goal}")$, which leads to a set of candidate concepts C_i^{cand} ;
 - (b) selecting the "best" candidate c_i out of C_i^{cand} ;
 - (c) selecting a given number of elements out of $pex^i_{c_i}$, thus creating p_i ;
 - (d) performing $CreateNegEx(c_i)$ to produce a given number of (good) negative examples for c_i , which we call the set n_i ;
 - (e) performing $ReplyQuery(path(c_i), p_i, n_i)$.
3. Ag_L collects the answers $(path(c_i), p_i, n_i)$ from all peer agents and uses a learner to learn c_{goal} from these combined examples (action $Learn((p_1, n_1), \dots, (p_m, n_m))$). If there are conflicts, then it resolves them with the help of the other agents using $AskClassify$ (resp. $ClassifyEx$ and $ReplyClass$ by the other agents).
4. Ag_L uses the learned c_{goal} and the collected $path(c_i)$ s from the other agents to construct an ontology path C_{path} leading to c_{goal} within its ontology O_L : (action $Integrate(path(c_1), \dots, path(c_m))$).

The result of this learning/teaching scheme is the description of c_{goal} in terms of Ag_L 's feature set F_L and an updated ontology $O_L^{new} = (C_L^{new}, \leq C, R_L, \sigma_L, \leq R_L)$. Ag_L will also create a set $pex^{c_{goal}}_L$ in case another agent wants Ag_L to teach it c_{goal} . Details of each step are explained in [1].

IV. ADJUDICATOR METHODOLOGY

The Adjudicator method aims at identifying and resolving conflict among peer agents (Ag_i s) regarding a learnt concept (c_{goal}) using a sound procedure to verify whether the differences in the viewpoints of the peers are statistically significant or not. Figure 1 shows the flow of the method.

In the Adjudicator method, after a successful learning cycle, objects representing the learnt concept are sent back to the peer agents asking for their feedback in the form of a two dimensional rate and weight data. This is after a careful consideration of parameters such as the number of peers and number of replicas. Therefore each query is backed up by a *statistical experiment*. Then response collected from peer agents is tested for integrity and analyzed using analysis of variance (ANOVA) for each dimension of the concept to identify whether a collective statistical consensus exists among the peer agent with respect to that particular dimension of the learnt concept. The same process will be repeated for all dimensions and the goal is to seek consensus for all dimensions.

In order to obtain statistically valid results, besides specifying the evaluation data, we require:

- (1) Selecting a proper number of agents to be asked for their feedback. For instance, asking 12 peer agents.

- (2) Collecting several data sets (i.e. replicas) from the queried agents, for example selecting to have at least 4 data sets for each feature (or dimension) examined.
- (3) Deciding upon the appropriate experiment type, for example, balanced incomplete block design (BIBD) [8] [9].

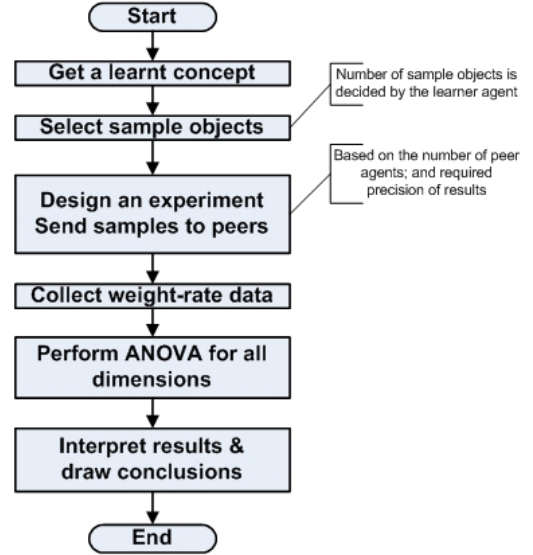


Figure 1. Flow of the Adjudicator method

Below we will give answer to the following questions: (1) What data should be collected from peer agents? (*Section A*) (2) How many objects should be selected and how many agents should be asked to provide feedback? In other words, what shall be the target population? (*Section B*) (3) How many replicas are needed? And what shall be the appropriate statistical model? (*Section C*). Finally, we present the details of statistical validation procedure (*Section D*).

A. Evaluation model and data collection

We have devised a *Multidimensional Weighted-Attributes Framework (MWAF)* based on which the peer agents can evaluate the example objects. The idea of *MWAF* is to use the most common and important criteria (or dimensions) and their features (or attributes) of an example for a concept being evaluated.

Dimensions: the framework contains a number of dimensions, each of which represents one of the major evaluation criteria.

Attributes: are the different features pertaining to each criterion (i.e. dimension) to describe it.

Parameters: the values that are given to measure the attributes.

The reason of partitioning features into dimensions is twofold. First, it allows for hierarchical decomposition of the concept; and second, it may show common grounds for conflicts. It is possible that the all peer agent agree on certain features and disagree on a few. Analysis of dimensions will reveal which features are commonly agreed upon and which are not.

The peer agents are asked to provide two parameters for each of the features: a *rate* and a *weight*. Rate reflects the extent the feature is present in the concept. For example, whether *color* is a feature for the concept *citrus* or not. Rate is an objective parameter because it is measured according to the degree of availability or effectiveness of the examined property. Weight, on the other hand, is a subjective parameter that reflects the extent that the peer agent thinks this feature is important to the concept being evaluated. For example, whether *color* is a necessary feature for the concept *citrus* or not. The values given to these two parameters can be binary. However, for the sake of generality of the method and providing better precision we have assumed them to be numeric with the range from 0 to 10. A value of ‘0’ implies full absence of the attribute, whereas a value of 10 reflects its maximum availability and strength.

Note that all available voting mechanisms merely collect data for the rate only. A unique feature of our method is to combine the rates and weights in order to differentiate between the *must have* and *nice to have* features of a concept.

B. Identifying sample set

Kitchenham et al [7] define a target population as the groups or individuals to whom the final results are applicable. In this work we can identify a two dimensional population: (1) a set of treatments (i.e. objects) that represent a concept; (2) and a set of peer agents Ag_s s that are queried.

Ideally, a set of objects which can be evaluated by Ag_s s to provide a valid set of observations are viewed as a representative subset for the learnt concept. The term “representative” is very critical, because if we do not have a representative set, we cannot claim that the final results can be generalized. Therefore identifying a proper subset of objects that represent the concept is crucial. In this work we assume that the representative sample set is selected based on the learner agent’s interest, and not at random from a large number of objects. In fact, this set can be selected by identifying objects that might have caused problem during the learning phase (pessimistic approach) or the objects that are best fit for the learnt concept c_{goal} (optimistic approach).

The set of peer agents includes but may not be limited to those who were queried at the first place. The minimum number of agents to be queried depends on the type of experiment (see below) and the number of replicas.

C. Selecting variables and statistical model

Before selecting the model, or even setting up the statistical hypotheses that describe the goal, it is important to define the experimental variables and the appropriate scale of measurement [4]. In our study, the effectiveness of the features, as described by the weights and rates given to each dimension, is the dependent variable (i.e. response). In order to be tested, a dependent variable is usually quantitative and measurable. The objects and peer agents are on the other hand independent variables that influence and regulate the

response; these variables are discrete in their nature and work through a nominal scale (categorical). When applying analysis of variance models, the independent variables are usually called factors or treatments.

For the sake of statistical validity each query should be tied to an appropriately designed experiment. There are several ways to construct an experiment including complete randomized design (CRD) [10], randomized complete block design (RCBD) [10] [3], balanced incomplete block design (BIBD) [8] [9]. The experiment design is decided upon based on (a) number of objects queried; (b) number of peer agents; and (c) number of replicas required.

D. Statistical validation procedure

For the statistical procedure we use the analysis of variance (ANOVA) model which is a robust and adaptable technique that is usually used to study the relationship between a dependent variable and one or more independent variables [5]. By using ANOVA, we can also identify the sources of variability among one or more potential sources. The basic underlying idea of ANOVA is to compare the variability of the observations between groups to the variability within groups. In Adjudicator method, ANOVA is used to test for the significant differences in the mean effectiveness of each dimension, D_m , among the example objects, such that:

- *Null hypothesis (H_0)*
There is no significant difference in the mean effectiveness of the examined dimension among the evaluated objects, i.e., all the treatment effects are the same and a consensus is achieved for this particular dimension.
- *Alternative hypothesis (H_a)*
There is a significant difference in the mean effectiveness of the examined dimension among the evaluated objects, i.e., the treatment effects are not the same and no consensus exists for this particular dimension.

ANOVA depends on the weight-rate data collected from the peer agents and in order to trust the results obtained from the ANOVA we shall check the model for aptness by running three tests, indicated in Table 1, to examine the assumptions involving the adequacy of the ANOVA procedure and to detect serious departures from the conditions assumed by the model [9]. Failure of any of the tests indicates that the collected data is not appropriate to validate the hypotheses.

Table 1. Model aptness tests

Test #	Test Type	Instrument Used
1	Outliers	a. Normal probability plot of residuals b. Individual value plot of residuals versus independent variable
2	Normality of residuals	Normal probability plot of residuals
3	Homogeneity of variances	a. Residual plots against fitted values b. Bartlett’s test

The validation process is repeated for all dimensions and

we can only assume a full consensus among the peer agents exist if the H_o holds for all dimensions. Unfortunately in many cases, it is hard to achieve such consensus. If the test result is significant, we can go further to carry out multiple pairwise comparisons of the objects, using Tukey’s HSD (honestly significant differences) method [14] to identify which pairs of objects are significantly different from the others. In this way we can pin point the candidate objects that are causing problem and possibly remove them.

V. EXAMPLE

In this example we created a test-bed and examined building conceptualization for 3 sets of software projects. We organized the entire projects documents in 3 repositories, one for each set. The sets were comprised of 110 software projects developed by software engineering students over 4 years period in the graduate and undergraduate courses. The sets were comprised of projects with concentration on software testing; concentration on object oriented analysis and design; and concentration on agent-based software development. We devised 12 agents, to serve as Ag_i peer agents, each assigned to a set of projects for one concentration in one year. We also designed an Ag_L learner agent by assigning to it random selection of projects from all three repositories. The conceptualization for each agent was arbitrary and they only shared a subset of features F and common concepts C as specified in Section III.

The learner agent Ag_L used the learning mechanism (Section III) to learn a new concept that we call it “development methodology”.

The learner agent Ag_L wants to verify whether a consensus exists among 12 peers (Ag_i s) queried with respect to this concept, using the Adjudicator method. The set pex^{goal}_L includes 9 objects that are all selected to be queried. The learnt concept has 3 dimensions (i.e., *modeling*, *process*, and *application*) and 14 features. In this experiment we decided to have at least 4 replicates which means that 4 data sets are collected for each of the 9 objects.

Giving this input set, for one-way ANOVA procedure using a CRD model requires that $i=36$ which means that we many need to query 36 peer agents so that each peer will receive one object at random to evaluate it, and each object will be evaluated by four peers. However, we have two limitations. We believe that there is heterogeneity among peers for many reasons that can potentially contribute to creating some sort of variability in data. Also another constraint is that we have only 12 peers. One way to overcome this lack is to make use of each peer to assess more than one object. This implies using a Randomized Complete Block Design (RCBD) where each agent should assess a complete block, i.e. 9 objects, or using the Balanced Incomplete Block Design (BIBD) model [14] in which a subset of objects (3 in this case) will be sent to each peer. In this experiment we decided to go for BIBD and query the peer agents to provide the evaluation data for 3 objects each.

In order to determine whether significant differences exist between the peers’ viewpoints with respect to objects, we repeated the experiment for the 3 dimensions that characterize the 9 objects. We could classify all 14 features into one dimension but we decided to cluster them into 3 dimensions each representing a specific criterion. The following set of hypotheses describes this strategy in a statistical fashion that will be applied to all the examined dimensions.

- **Null hypothesis, H_o :** There is no significant difference in the mean effectiveness of the examined dimension among the evaluated objects.
- **Alternative hypothesis, H_a :** There is a significant difference in the mean effectiveness.

We analyzed the collected data by means of applying ANOVA procedure to the BIBD model to test the significant differences in the mean effectiveness of each individual dimension. In this way, if significant differences are ascertained, we go further to perform pairwise comparisons to identify which pairs of objects significantly differ from which ones. On the other hand, if the overall ANOVA test is insignificant, we will not apply any pairwise comparisons. In such a case, the conclusion to be made is that all the objects are statistically equal in their main effects for the examined dimension. The detailed steps of analysis were:

Step 1: Data Abstraction and Formulation

We extract the necessary data for this dimension from the collected raw data. Thus, we multiply recorded rates by the corresponding average weights of relevant attributes.

Step 2: Constructing the BIBD tableau

By adopting the BIBD arrangement we construct the BIBD tableau of this dimension.

Step 3: Testing the ANOVA assumptions

We conducted the tests depicted in Table 1 to examine the assumptions involving the adequacy of the ANOVA procedure. The results of these tests were that all of the plots did not suggest any significant departures, either from the normality of the distribution of errors or the homogeneity of error variances. Also, there was no evidence of potential outliers; all the residuals appeared to be bounded within a 95% confidence interval, and they all fell within the acceptable range of normality [9].

Step 4: ANOVA computations and hypotheses testing

We applied the adjusted formulas of the BIBD described by Yates [14], to the data arranged in the BIBD tableau and came up with the analysis of variance components. According to [14], if the calculated F statistic (F_0) is larger than its critical value (F_{crit}), F_0 falls in the rejection region. Thus, we have sufficient evidence to reject the null hypothesis at a 95% level of significance based on the available data. In this example F_0 was larger than F_{crit} for 2 out of 3 dimensions, i.e., *modeling* and *application*.

Step 5: Identifying significant differences

In the hypotheses test we carried out in Step 4, no variability was observed for the Dimension 2 (*process*) therefore no need to continue for this dimension. However, since the test was significant for the other 2 dimensions, we

went further for pairwise comparison tests to identify which objects are statically different. *Christensen* [2] and *Neter* [10] have exhibited several methods for pairwise comparisons, such as *Tukey's HSD* (Honestly Significant Differences, also known as the *T*-method) and *Fisher's LSD*. In this case, we adopted the HSD method and determined that the pairs of objects which have significantly different effects were: [O1] with [O3, O4, O5, O6, O7, O8]; [O2] with [O5, O6]; and the rest were not significantly different.

Table 2 shows a binary representation of these results. It should be noted that an intersection of '0' in a cell implies that the corresponding two objects, as crossed by their row and column, are **not** significantly different. That is, both objects are equivalent statistically, although they may have different means of effectiveness. On the other hand, a value of '1' implies that the two objects are significantly different. This clearly indicates that for Dimension 1 (*modeling*) the objects O1 and partially O2 are causing trouble and there exists considerable consensus on the rest. By further reviewing the objects we found out that the O1 and O2 were instances of two popular agent based development methodologies that use rather different modeling perspective that the rest which were examples of the conventional object oriented approaches.

It may look rather odd to find that for Dimension 3 (*application*) the overall test conducted by the ANOVA was significant, while the pairwise comparisons of means conducted by the HSD test fails to reveal any significant differences among the objects in this dimension. This exceptional case occurs because the ANOVA simultaneously considers all possible contrasts involving the treatment means, and not just the pairwise comparisons. From the agent perspective, this case implies that the peer agents could not reveal significant differences among the object in the set of attributes described by Dimension 3.

Table 2: Binary representation of the evaluation results

		O2	O3	O4	O5	O6	O7	O8	O9
Dimension 1: Modeling	O1	0	1	1	1	1	1	1	1
	O2		0	0	1	1	0	0	0
	O3			0	0	0	0	0	0
	O4				0	0	0	0	0
	O5					0	0	0	0
	O6						0	0	0
	O7							0	0
	O8								0
		O2	O3	O4	O5	O6	O7	O8	O9
Dimension 3: Application	O1	0	0	0	0	0	0	0	0
	O2		0	0	0	0	0	0	0
	O3			0	0	0	0	0	0
	O4				0	0	0	0	0
	O5					0	0	0	0
	O6						0	0	0
	O7							0	0
	O8								0

Step 6: Interpretation of results

Learner agent could identify the cause for discrepancy among peers. Among 14 features and 3 dimensions only 2

were the cause of trouble and pairwise analysis revealed that two objects O1 and O2 may be removed from the learned examples set. The learning (Section III) by discarding these two objects and adjudicator method (Section IV) may be run again until no further discrepancies are found.

VI. CONCLUSION

We presented a methodology for improving interaction and communication among agents by letting them learn ontological concepts from each other while maintaining their own individualized conceptualization. The Adjudicator method presented in this paper provides the learner agent with a conflict resolution mechanism that goes beyond a simple voting and helps achieve consensus among peer agents with regard to a learnt concept. Therefore the Adjudicator method replaces ad-hoc voting methods and is reliable because it adopts statistical procedures and validates the concluded results within certain confidence limits.

ACKNOWLEDGMENT

The authors are grateful to NSERC for the Discovery Research Grant and IBM for the 2006 UIMA Innovation Award.

REFERENCES

- [1] M. Afsharchi and B.H. Far and J. Denzinger: Ontology Guided Learning to Improve Communication among Groups of Agents, Proc. 5th AAMAS 2006, 2006, pp. 923-930.
- [2] R. Christensen: Analysis of Variance and Regression. Chapman & Hall, 1996.
- [3] L. Douglass, BIOM 602: Lecture Notes, Oregon State Univ., 2004.
- [4] A.H. Elamy and B.H. Far: Utilizing Incomplete Block Designs in Evaluating Agent-Oriented Software Engineering Methodologies. In Proceedings of the IEEE CCECE'05, Saskatoon, Canada, May 2005, 1412-1515.
- [5] R. Fisher: The Design of Experiments, 1st ed., Oliver & Boyd, Edinburgh, 1935.
- [6] K-C. Jim, C.L. Giles: Talking Helps: Evolving Communicating Agents for the Predator-Prey Pursuit Problem, Artificial Life 6(3), 2000, pp. 237-254.
- [7] B. Kitchenham, and S. Pfleeger: Principles of Survey Research, Part 5: Populations and Samples. Software Engineering Notes, vol. 27, no. 5, UK, 2002.
- [8] M. Liu and L. Chan: Uniformity of Incomplete Block Designs, Int'l Journal Materials and Product Technology, vol. 20, no. 1-3, 2004, pp.143-149.
- [9] D. Montgomery: Design and Analysis of Experiments, 6th ed., John Wiley & Sons, Inc., USA, 2005.
- [10] J. Neter, W. Wasserman, and M. Kutner: Applied Linear Statistical Models, 5th ed., Irwin, USA, 1996.
- [11] L. Steels: The Origins of Ontologies and Communication Conventions in Multi-Agent Systems, Autonomous Agents and Multi-Agent Systems 1(2), 1998, pp. 169-194.
- [12] G. Stumme: Using Ontologies and Formal Concept Analysis for Organizing Business Knowledge, in J. Becker, R. Knackstedt (Eds.): Wissensmanagement mit Referenzmodellen – Konzepte für die Anwendungssystem- und Organisationsgestaltung, Physica, 2002, pp. 163-174.
- [13] A.B. Williams: Learning to Share Meaning in a Multi Agent System, Autonomous Agents and Multi Agent Systems 8(2), 2004, pp. 165-193.
- [14] F. Yates: Experimental Design: Selected Papers. Griffin, UK, 1970.