First Steps in Context Modeling for Conflicts Characterization in Cooperative Task-Execution Support Systems

Michel MUNOZ

IRIT − CSC
UPS, 118 route de Narbonne − F 31062 Toulouse Cedex 4
munoz@irit.fr

Abstract. The help provided by task execution support systems must be pertinent, cooperative, and robust. Such systems cannot qualify or quantify a given situation's cooperation-level owing to the lack of suitable theoretical framework. In this paper we present first elements to define this framework. Such a framework would bring several benefits. First, it would allow systems to assess the cooperativity/pertinence of their actions by comparing current situation with the situation resulting from doing a given action. Next, by detecting and describing problems of current situation it could give the system clear motivations and justifications for intervening. Finally, characterizing possible conflicts in a domain-independent way would allow us to link each class of problem to problem-solving strategies hence giving the system some predetermined conflict-solving behaviour. Each of these aspects contributes to the quality of the help provided by systems.

1 Introduction

Task execution support systems (Babaian and al. 2002; Rich, Sidner and Lesh 2001; Ferguson and Allen 1998) are seen as agents involved in the activity. This kind of system relies heavily on models i.e. domain models, user model, task models, cooperation model, etc. (see for example (Soubie 2003; Flycht-Eriksson 1999)).

This work in progress is mainly targeted at Cooperative Knowledge-Based Systems (CKBS, see (Soubie 1996)) but the approach and results can be easily reused in other support systems. CKBS targeted domains are those where activity strongly depends on contextual aspects (events, state, abilities of agents, etc.), i.e. CKBS is suitable in situations where context can cause critical problems for task accomplishment. Problems combinatorics is then so large that the system must be inferential (e.g. dynamic management of cooperation, activity, constraints, etc.). The provided help has to be appropriate, cooperative, and robust i.e., the interventions of the system must take into account the context, user, activity, etc., must tolerate problems occurring during activity and must not engender new problems.

One problem with systems presented so far is the lack of pragmatic framework allowing support system to:

- evaluate to which extent a given situation is cooperative
- justify why it choose a given problem solving strategy
- prove why and how much its intervention is cooperative and pertinent

Moreover, describing non-cooperative situations in a formal way and linking those descriptions to problem-solving strategies would enrich its interventions strategies hence augmenting help capability of the systems.

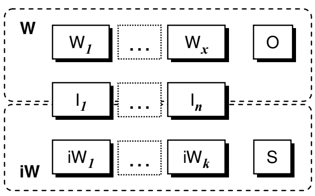
The main difficulty of providing such a framework lies in the wide range of domains where systems can be used. We will limit our work to cooperative activity situations, that is, situation where agents work together for a particular purpose. Another difficulty is the interweaving of different levels of activity within the global activity, e.g. role management, communication, tasks, social constraints, etc; the problem here is in having a homogeneous theoretical − and practical − approach of this diversity.

In this paper we present first elements from a work in progress intending to define a framework dealing with this problem.

2 Situation Modeling

In this section we describe how an agent sees an activity situation. This description is domain-independent and can describe the variety of activity situations that support systems may encounter. It is the ontological commitment of our future work. Stated another way, this is how we see the global context, that is, all the situation.

World Modelling. The part of the world involved in the activity is divided up as follows:



The material world W is made up of physical entities, i.e. entities having spatial properties. The immaterial world iW is made up of volumeless entities such as files, computer programs, etc.
The parts of the world (W_i or iW_i) are structured in a hierarchy based on a *part-of* or *contains* relationship − an oriented acyclic graph. Yet this last point is not visible on the drawing for clarity.

The agents are O − the operator − and S − the system. Both are seen as a special case of a part of the world. Even if we use only one S and only one O, the model can extend to more operators and systems.

The means of interaction I_j contain items covering many realities. These items can be distinguished by considering where inputs come from and where outputs are going:

- $W \rightarrow W$, e.g. communication code, natural langage…
- $W \rightarrow iW$, e.g. keyboard, sensors, vocal command…
- $iW \rightarrow W$, e.g. screen, printer, computer-driven furniture…

- iW ↔ W, e.g. device with feedback (in a flight simulator : the simulated reality iW, constraint the behaviour of the cockpit W and vice-versa, i.e. W lead the behaviour of iW)
- iW → iW, ex. GUI, running program, network protocol…

Each of these elements has a synchronic description (current state) and a diachronic description, i.e. an historical record of its state.

Some problems are specific to groups of agents, i.e. roles, cooperation, social rules. In order to account for them we introduce the concept of group G. A group contains a definition, namely agents and roles involved, tasks, contextual elements. If this definition holds in a given situation then the system will assume − until proved otherwise − that he observes an instance of this group.

To each of aforesaid concepts − S, O, W, iW, G − we associate some of the following facets:

- Intentions (Int): one agent's intentions which are explicitly shown or (easily) deduced from related knowledge (e.g. task models). These intentions are described by objects containing their characteristics: intended effects, constraints, agents involved, related tasks, etc.
- Abilities (Abl): recipes, plans, task models. A given ability may rely on other abilities, resource, means, etc.
- Rules (Rul): definitions, conventions, constraints, etc., any set of rules that are shared and used by groups of agents, most of the time, the rules come from organisation or conventions.
- Knowledge/Beliefs (KB): knowledge and beliefs explicitly shown by an agent or (easily) deduced from related knowledge (e.g. task models, user models).
- State (Stat): current state of one entity (e.g. existence, physical state, etc.)
- Resources (Res): resources, time… elements consumed during activities.
- Means (Mn): means, tools, physical space… non-consumable elements needed during activities.

By combining elements presented so far we obtain the situation categories described in table 1. In a given situation, every observed element − every information observed during interaction − must belong to only one of these categories.

Up to now, we have elements allowing to describe entities and information of a given situation. Relating entities − agents, tools…− to table 1 entries is done at modelling time. The problem is to relate observed interaction data to table 1 entries at run time. In order to do so, we need a notion of "context of interpretation", i.e. a local context − a contEXT − limiting possible meaning of what is observed. This is not to confuse with the global context which is only the situation, i.e. the state of the world.

Context modeling. In order to allow inputs interpretation and to relate observed data to information categories we define an extended context (contEXT) concept. A contEXT is composed of: one definition, data gathered by observing the activity, and a summary of these data.

A data is related to a contEXT if the current (global) context respects the contEXT definition, e.g. the presence of certain agents, properties of contextual elements, current activity, dialogue topic… Therefore a contEXT is a group of observations consistent with the contEXT definition.

A context can have a more or less abstract summary of the data it contains. The motivation behind this summary is to allow the system to reason on less data and domain-independent data, i.e. reasoning about cooperation using an abstract description of the current situation.

There are several levels of contEXT; a contEXT contains either "atomic" data or other contEXTs. The contEXTs' structure is a directed acyclic graph. The more a contEXT is high in the hierarchy the more is its summary abstract.

In order to better grasp this concept let's go through some examples. Suppose we define contEXTs only by tasks, i.e. every time a particular task start, a new instance of the related contEXT is created. For example, each time a DiagnoseRepair joint activity `start, a` DiagnoseRepairContEXT is created. Now, since this contEXT is linked to the elements in its definition we can interpret ongoing interactions using active contEXTs. If, for example, an agents starts using some electronic measuring device, we could know − using models − that this tool is used in many activities among which Diagnose, so the system can interpret the action of the agent as "I started to diagnose the problem". This, allow the system to infer more elements: agent's intention (do Diagnose), agent's state (if agent do diagnose then his hands are occupied), agent's beliefs (agents believes he has the capability to carry out the task)…

Of course, at a given point during the activity there are multiple open contEXTs hence the necessity of structuring them. In the above example you may imagine that during the diagnostic activity an information-seeking dialog starts between the system and the operator. In this case, the InfoSeekContEXT will be a sub context of DiagnoseRepairContEXT. With such a hierarchy, observed interactions will first be interpreted in the more specific context − InfoSeekContEXT − and if it is not possible, the system will try to use immediate super context − DiagnoseRepairContEXT.

In the example, the "summary" aspect of the context may be "Agent X started to diagnose on his own then he asked for information about Y…" This gives the system a synthetic view of the situation without all the details that led to these events − sentences, interactions, interruptions, …

3 Envisioned Approach for Situation Characterization

From now on we describe how we envision future research.

Principles. The main idea is to have a language that allows describing the situation in a form that is both synthetic and domain independent and then to reason using that description. Furthermore we aim to link problem solving strategies to each class of problem. Such approach would allow us to add a cooperativity reasoning module and a problem solving capability to support systems. In addition, since those systems rely heavily on models then providing a prebuilt library of problem solving strategies would lessen the cost of modeling a particular domain.

A situation or a part of a situation – a given contEXT – will be described by the conflicts/problems being present in it. A situation is cooperative if it has no conflicts or if all conflicts are being fixed up.

A conflict is an incompatibility between two elements of situation categories, e.g. between an agent's intention and the state of a part of the world.

An intervention is cooperative if it contributes to solve one or more conflicts. An intervention is pertinent if it doesn't create new conflicts, be it in a direct or indirect way.

The best intervention is the one which solves the more conflicts or the most critical conflicts, and at the same time creates as little conflicts as possible. It is then the most cooperative and pertinent strategy.

Approach. We will try to use the less formal vocabulary as possible; the difficulty being to find a simple language to describe conflicts. Up to now we use: ⇉ (compatibility), ⇆ (incompatibility, conflict), **!** (intention to do something), **!!** (intention that something holds), **:** (sequentiality), **=** (equality). For example "**⇆ : !A⇉**" means that at some time a conflict has been observed, and later (sequentiality) the conflict has been intentionally fixed by agent A, for example by means of an action, a change of intention, etc.

An *atomic* conflict is only defined by the elements being incompatible; a non-atomic conflict is one defined by other conflicts, i.e. it is a conflictual situation containing many conflicts.

We are still exploring the issue of knowing whether it is possible to list and describe all possible conflicts *a priori*, and if it is possible, to which extent it is so.
The original naïve approach consisted in doing a Cartesian product of the nineteen categories of table 1 hence giving 361 possibilities, each of these containing several atomic conflicts; for example, A_i being an agent, in the "A_1.KB×A_2.KB" case, "⇆" means mutual misunderstanding, and "!⇆" means disagreement. The problem is that all the possibilities do not have a meaning, for example the "W_1.Stat×A_3.Stat" has no meaning, such a conflict is impossible by definition. Conversely, an heterogeneous conflict may have a meaning, e.g. an agent A wants B to do something (call it intention n°15) but organizational rules don't allow it because of the relative social status of A and B (call it rule of group n°12); this would be formalized as "A.Int_{15}×G_{12}.Rul ⇆". We are exploring if there are criteria for meaningful combinations.

There are many other open issues among which:

- From which level of context could we add/compute a summary and from wich level of summary could we try to detect meaningful conflicts.
- Is the concept of contEXT powerful enough to group data from observation or is an other mechanism needed
- To which extent is it possible to link conflict classes to problem-solving strategies (Reed and Long 1997)
- The use of cooperative systems implies that an intervention may mix action and communication. The problem is then twofold: which dialogue ontology do we have to use to describe communication? (Kinds of dialogue, dialogue games, communication acts, etc. (Clark and Popescu-Belis 2004; Traum 2000; Mann 2002))? How can we describe dialogue and action in a simple and homogeneous way? (e.g. (Mateas and Stern 2002))
- If a contEXT has several sub-contEXTs: how can the sub-contexts be combined?

4 Possible uses of conflict reasoning

Case n°1: choosing intervention according to induced conflicts. The domain is maintenance and repair. The situation is as follows: O is doing some maintenance at a distance from the computer "containing" S; the only means of interaction are one graphical user interface and the capacity of emitting a beep; both are located on the computer. At a given moment, the organization (Org) tells S to inform O about something. S computes that only two ways of communicating are available: (1) calling out (i.e. beeping) O and displaying the message on the computer screen, i.e. instant communication; (2) displaying the message on the computer screen without calling O, hence waiting O to come back to the computer, i.e. delayed communication.

Choice 1 implies O's activity interruption, i.e. if O hears the beep he will go to computer to check the reason of S call. Choice 2 delays communication accomplishment but has the advantage of not interrupting O.

Now, let's say that two groups are defined and suitable for this event. One global group is suitable all the time − e.g. "agent inside organisation" − and this group states that Org has priority over any operator or activity (rule 1). The other group can be applied when the activity is maintenance; this group states − group's rules − that is better not to interrupt an operator while is working (rule 2).

If S chooses choice 1, it will cause a conflict between the intention of Org (through S), namely to tell O something, O's intention who wants to finish his task, and rule 2. If S chooses choice 2, it will cause a conflict between rule 1 and the need to wait until O finishes his task before having the possibility to communicate the information.

In both case we have conflicts, but considering that rule 1 (obligation) has priority over rule 2 (advice), the first choice is the best one since it implies the least serious effects.

Case n°2, maximizing intervention pertinence. The domain is rail travel information. An agent A needs a piece of information (note that this situation could be described as an A.Int×A.KB conflict).

A asks S − e.g. an information kiosk − if there is a train going to Paris. S makes the hypothesis that A wants to take a train for Paris but does not have enough information to do so; therefore there is a conflict − incompatibility, inadequacy − between A.Int and $A.KB_1$, $A.KB_2$…as much knowledge as needed for knowing if such a train exist and − if necessary − to carry out a "take a train" action.

One strategy may be to give a yes/no reply, hence solving one conflict ("Yes there is such a train"); another strategy is to give as much information as necessary in order to take the train. The second option is more cooperative than the first because it solves more conflicts[1].

Case n°3, detecting end of (sub-)dialogues. Two agents A and S are having a conflict of intentions. In order to resolve this conflict they entered a negotiation dialogue. At one moment during this dialogue, A starts an information-seeking sub-dialogue (A wants to know something to take a decision). Later during that subdialogue, A says "well, I'll do Y". S then tries to understand the side effects of that locution using current contexts, i.e. the context of negotiation dialogue and the one of information-seeking. The statement of A has little meaning in current context (A seeking information from S) but may make sense in the immediately enclosing context (negotiation about intentions). S sees that the new intention of A − i.e. Y − is compatible with his own intention, so the initial conflict is solved. S can take for granted that the conflict is solved and therefore that the two dialogues are closed (since all dialogues boil down to solving the initial intention conflict) even if there are no explicit marker of the end of the dialogues. An other option for S is to check its deduction by opening a short confirmation dialogue, e.g. "We do not have problem anymore, do we?" A positive answer from A would imply the closing of the three currently open dialogues, i.e. negotiation, information, confirmation.

5 Conclusion

This paper presented elements of an ongoing work dealing with the generic characterization of conflict in cooperative situations. Such an approach is new and its main benefits are: a characterization of the quality of the cooperation in a given situation, a motivation for the system to intervene, and a mean to anticipate − at least to some extent − if and how much an intended action is cooperative and/or pertinent. All these elements are uniformly described since everything − even communication − is seen as an action.

[1] Of course such a conclusion is simplistic but the example sketches the principle used.

This work is still at an early stage. Upcoming work will deepen the notions of atomic conflicts and extended contexts. More specifically the next problem to tackle is to list possible conflicts and to characterize them *a priori*.

Table 1. Information categories for cooperative activity situations.

①	②	③
World **W_h, W_k**	Rul	Factual definitions (e.g. « a penguin cannot fly ») Process definitions : Law (ex. « if one Agent grabs an Object then the object will move with the agent »), Automata (e.g. state-transition describing some (i)W_k)
	Stat	Set of entities pertaining to the (i)W_k part of the universe.
Mean of interaction **I_j**	Abl	Abilities: e.g. « catch attention », kind of communicative act, kind of dialogue/activity…
	Rul	Factual definitions (e.g. uses such modalities, relies on such communication code…) Process definitions (e.g. Conventions, Codes, Habits,state-transition…)
	Stat	Current state of the I_j
	Res	e.g. paper for a given printer
	Mn	Necessary means for I_j's availability (e.g. to have a given amount of display area), i.e. the things you need so that I_j works. The means necessary so that an available I_j is also usable (e.g. in order to use a mouse you need a planar surface, such vocal command input is unusable above a given level of noise, to use such device an Agent must have such and such ability or characteristic).
Agents **O, S**	Int	Intentions
	Abl	Plans, recipes, etc. known beforehand ou expressed (e.g. an agent stated that he knows how to do X while system ignored it)
	Rul	Rules that are independent of organisation (e.g. psychological side-effects of certain physical states). Socio-organisational rules (e.g. in a given context agent A can give orders to every other agent having the rank of X, Y, Z…)
	KB	Knowledge, beliefs
	Stat	Physical, emotional, cognitive,… state (e.g. say O is panicking, so – O.Rul – being cognitively « freezed ». S cannot start a next-priority-negociation-dialogue with O, if social rules are applicable to the current situation then S can take the situation in hand)
	Mn	Activity and/or domain relevant physical characteristics
Groups **G_m**	Int	Joint intentions of involved agents, and intention of the group by itself (e.g. carrying out a task)
	Abl	Plans of the group
	Rul	Group's rules, namely rules specific to a given group, indicating how one is expected to behave inside the group. These rules may have a double use, i.e. normative – behaviour constraint– and descriptive – allowing to predict probable future behaviours.
	KB	Group's shared knowledge, e.g. agents' commitments
	Stat	Activation status of the group (e.g. group is activated, suspended, ended, etc.)
	Mn	Necessary means for G_m existence and « use » (e.g. to have such hand such roles, such contextual elements)

① = Part of the universe, ② = Facet, ③ = Meaning

References

Babaian T., Grosz B., Shieber J., Stuart M., (2002). A Writer's Collaborative Aid, *Proceedings of the Intelligent User Interfaces Conference*, San Francisco, CA. January 13–16. ACM Press, pp. 7–14.

Rich C., Sidner C., Lesh N., (2001). COLLAGEN Applying Collaborative Discourse Theory to Human-Computer Interaction, *AI Magazine*, Vol 22, No 4, 15–25.

Ferguson, G., Allen, J. (1998). TRIPS: An integrated intelligent problem-solving assistant. *Proceedings of the Fifthteenth National Conference on Artificial Intelligence*, pp. 567–572). Menlo Park, CA: AAAI Press.

Soubie J.-L., (2003). On the role of multi-dimensional models in man-machine cooperation, *Revue d'intelligence artificielle*, V. 16, N. 4-5/2002, pp. 545–559.

Flycht-Eriksson, A. (1999). A Survey of Knowledge Sources in Dialogue Systems. In: *Proc. of IJCAI'99 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*. Stockholm, Sweden, pp. 41–48.

Soubie, J.-L., (1996). Coopération et systèmes à base de connaissances, *Habilitation à diriger des recherches*, Université Paul Sabatier, Toulouse.

Reed C.A., Long D.P., (1997). Collaboration, Cooperation, and Dialogue Classification, *IJCAI'97 Workshop on Collaboration, Cooperation, and Conflict in Dialogue Systems*.

Clark A., Popescu-Belis A, (2004). Multi-level Dialogue Act Tags, *SIGDIAL'04* (5th SIGdial Workshop on Discourse and Dialogue), Cambridge, MA, USA, 163–170.

Traum D., R., (2000). 20 Questions for Dialogue Act Taxonomies, in *Journal of Semantics*, 17(1):7–30.

Mann, W. C. (2002). Dialogue Macrogame Theory. SIGdial Workshop, *Proceedings of the Third SIGdial Workshop on Discourse and Dialogue*, ACM, 129–141.

Mateas, M., Stern, A. (2002). A Behavior Language for Story-based Believable Agents, *Papers IEEE Intelligent Systems issue on AI and Interactive Entertainment*, Vol 17 No 4. 39–47.