Mephu Nguifo E., Dillenbourg P. & Baker M (1999). A comparison of learning and dialogue operators for computational models. In P. Dillenbourg (Ed) *Collaborative-learning: Cognitive and Computational Approaches*. (pp. 122-146). Oxford: Elsevier

Chapter 7

Knowledge transformations in agents and interactions :

a comparison of machine learning and dialogue operators

Mephu Nguifo E
CRIL - IUT de Lens, France
mephu@cril.univ-artois.fr

Baker M.J.
CNRS, France
mbaker@univ-lyon2.fr

Dillenbourg P.
TECFA, Switzerland
Pierre.Dillenbourg@tecfa.unige.ch

Abstract

This paper addresses the problem of understanding the mechanisms by which learning takes place as a result of collaboration between agents. We compare dialogue operators and machine learning operators with a view to understanding how the knowledge that is co-constructed in dialogue can be learned in an individual agent. Machine Learning operators make knowledge changes in a knowledge space; dialogue operators are used to represent the way in which knowledge can be co-constructed in dialogue. We describe the degree of overlap between both sets of operators, by applying learning operators to an example of dialogue. We review several differences between these two sets of operators: the number of agents, the coverage of strategical aspects and the distance between what one says or hears and what one knows. We discuss the interest of fusing dialogue and learning operators in the case of person-machine cooperative learning and multi-agent learning systems.

1. Introduction.

What is the specificity of collaborative learning with respect to learning alone? A well-known answer is for instance sharing the cognitive load, a mechanisms which can be easily be translated into a multi-agent architecture. But, the more fundamental point is that, during collaborative learning, different types of interactions occur between learners, some of which have specific cognitive effects. Therefore, a computational model of collaborative learning should integrate the mechanisms which relate interactions with cognitive effects. These mechanisms cannot be a simple 'information flow' process, where Agent-A learns

X simply because Agent-B communicates X to Agent-A. Such a model would for instance contradict the facts that the explainer often gets more benefit than the explainee (Ploetzner & al, *this volume).* Hence, we are looking for models which deeply integrate dialogue and learning (Dillenbourg, 1996). Specifically, in this chapter we look for convergences between the basic operators isolated in machine learning (ML) research and in research on dialogue. Beyond this scientific goal, this work has also practical implications: to integrate dialogue operators into ML algorithms to adapt these algorithms, on one hand, to the interactions with a user (e.g. Mephu Nguifo, 1997), and on the other hand, to interactions with other artificial agents (learning in multi-agent systems).

Thousands of ML systems have been developed, clustered into categories such as "explanation based learning", "similarity based learning", "reinforcement learning", ... Can we describe this variety of algorithms with a restricted set of elementary learning operators ? Can we integrate them into multi-strategy learning systems? Michalski (1993) addressed these questions. He proposed a set of operators in knowledge space, termed *knowledge transmutations*, which cover the range of inferences mechanisms introduced in ML algorithms. Transmutations are generic patterns of knowledge change. A transmutation may change knowledge, derive new knowledge or perform manipulations on knowledge that do not change its content.

This search for "atoms" within ML is common to many scientific fields, and namely, in dialogue studies. Some researchers attempt to understand the mechanisms by which dialogue leads to learning. They studied specific types of interaction associated with learning, such as negotiation (of knowledge, of dialogue focus), argumentation and explanation. Here we restrict ourselves to considering "knowledge negotiation" or "co-construction of knowledge". In this case, the question arises as to whether we can describe knowledge negotiation in dialogue with an "atomic" set of operators. There exist many classifications of dialogue units, reflecting different theoretical approaches. Here we use the approach proposed by Baker (1994). This includes a set of operators, termed *transformation functions,* that describe — at the knowledge level — the relations between the contents of utterances produced in collaborative problem-solving dialogues. These operators can be compared with research on content (and other) relations between segments of texts, such as Hobbs' (1982) classification of "coherence relations" in discourse and Mann & Thompson's (1985) "rhetorical relations".[1]. There are also many other types of relations that exist between segments of dialogue, other than those that obtain purely on the knowledge level, such as functional relations between speech acts (see, e.g. Levinson, 1983 ; Moeschler, 1985). What is specific about the approach described in Baker (1994), and particularly adapted to our purposes here, is that it is concerned with relations in dialogue, rather than in text, and that it is based specifically on analysis of collaborative problem-solving interactions.

[1] See Sanders, Spooren & Noordman (1992) for a review and synthesis of different categories of textual relations.

This chapter compares the two above mentioned set of operators, Michalski's "transmutations" and Baker's "transformations". They have been chosen as particularly suitable examples, without implying that they are unanimously recognised in their respective scientific community as the common reference. We first describe briefly these classifications, respectively in section 2 for Michalski and 3 for Baker. We then attempt to apply these operators across disciplines, i.e. to use Michalski's machine-learning operators for describing knowledge negotiation in a dialogue, and to use Baker's dialogic operators on a well-known machine learning problem. This enables us to explore the extent to which knowledge transformations that take place within a single agent (i.e. machine learning) and during the interaction between two agents can be modelled in terms of similar processes. This leads (section 5) to a theoretical comparison between the two sets of operators. Finally, in the last section we draw more practical implications on the interoperability of dialogue and learning operators with respect to the goals stated above: modelling collaborative learning, and implementing human-machine collaborative learning systems.

2. A taxonomy of machine learning operators

Michalski (1993) defines learning as follows: *Given* an input knowledge (I), a goal (G), background knowledge (BK) and a set of transmutations (T), *determine* output knowledge (O) that satisfies the goal, by applying transmutations from the set T to input I and/or background knowledge BK. Transmutations change the knowledge space, i.e. the space where can be represented all possible inputs, all of the learner's background knowledge and all knowledge that the learner can generate. A transmutation may change existing knowledge, derive new knowledge or perform certain manipulations on knowledge that do not change its content.

To define these operators, Michalski introduces two concepts: a reference set and a descriptor. A *reference set* of statements is an entity or a set of entities that these statements describe or refer to. A *descriptor* is an attribute, a relation, or a transformation whose instantiation (value) is used to characterise the reference set or the individual entities in it. For example, consider a statement: “Paul is small, has a PhD in Computer Science from Montpellier university, and likes skiing”. The reference set here is the singleton “Paul”. The sentence uses three descriptors: a one-place attribute “height(person)”, a binary relation “likes(person, activity)” and a four-place relation “degree-received(person, degree, topic, university)”. The reference set and the descriptors are often fixed once and for all in a ML system.

Two categories of transmutations functions are defined:

- *Knowledge generation transmutations* change informational content of the input knowledge. They are performed on statements that have a truth status. These transmutations (see table 1) are generally based on deductive, inductive, and/or analogical inference.

- *Knowledge manipulation transmutations* are operators that view input knowledge as data or objects to be manipulated. There is no change of the informational content of pieces of knowledge that composed the whole knowledge. Examples are insertion/replication, deletion/destruction, sorting or unsorting operators.

Generalization extends the reference sets of input, i.e. it generates a description that characterizes a larger reference set than the input..	*Specialization* narrows the reference set of objects.
Abstraction reduces the amount of detail in a description of the given reference set.	*Concretion* generates additional details about the reference set.
Similization derives new knowledge about a reference set on the basis of the similarity between this set and another reference set about which the learner has more knowledge.	*Dissimilization* derives new knowledge on the basis of the lack of similarity between the compared reference sets.
Association determines a dependency between given entities or descriptions based on the observed facts and/or background knowledge. Dependency may be logical, causal, statistical, temporal, etc.	*Disassociation* asserts a lack of dependency. For example, determining that a given instance is not an example of some concept, is a disassociation transmutation
Selection is a transmutation that selects a subset from a set of entities (a set of knowledge components) that satisfies some criteria. For example, choosing a subset of relevant attributes from a set of candidates, or determining the most plausible hypothesis among a set of candidate hypotheses.	*Generation* generates entities of a given type. For example, generating an attribute to characterize a given entity, or creating an alternative hypothesis to the one already generated.
Agglomeration groups entities into larger units according to some goal criterion. If it also hypotheses that the larger units represent general patterns in data, then it is called clustering.	*Decomposition* splits a group (or a structure) of entities into subgroups according to some goal criterion.
Characterization determines a characteristic description of a given set of entities. For example, a simple form of such description is a list (or a conjunction) of all properties shared by the entities of the given set.	*Discrimination* determines a description that discriminates (distinguishes) the given set of entities from another set of entities.

Table 1: Pairs of opposite knowledge generation transmutations (Michalski, 1993)

Transmutations are bi-directional operations: they are grouped into pairs of opposite operators, except for derivation that span a range of transmutations.

Derivations are knowledge generation transmutations that derive one piece of knowledge from another piece of knowledge (based on some dependency between them), but do not fall into the special categories described above. Because the dependency between knowledge components can range from logical equivalence to random relationship, derivations can be classified on the basis of the strength of dependency into a wide range of forms.

- *Reformulation* transforms a segment of knowledge into a logically equivalent segment of knowledge.
- Deductive derivation, Abductive Explanation and Prediction can be viewed as *intermediate derivations*. A weak intermediate derivation is the cross-over operator in genetic algorithm (Goldberg, 1989). Mathematical or logical transformations of knowledge also represents forms of derivations.
- *Randomization* transforms one knowledge segment to another one by making random changes. For example, the mutation operation in a genetic algorithm (Goldberg, 1989).

In the following, we restrict ourselves to the first category (Table 1): changes at the knowledge level, which can later be compared to Baker's knowledge level operators. It is more difficult to relate operators which concern linguistic form since the form of an utterance is very different from the AI knowledge representation scheme.

3. A taxonomy of dialogue operators

A key issue in the study of collaborative problem-solving is to understand how problem-solutions are jointly produced in dialogue. Such common solutions can rarely be reduced to simple 'accumulations' of individually proposed solution elements. Rather, solutions *emerge* by an interactive process in which each agent (learner) *transforms* the contributions of the other, in order to attempt to arrive at a mutually satisfactory solution element. This process may be described as one by which *knowledge* is *co-constructed* by a process of *negotiation* (where the term 'knowledge' is relativised to the agents concerned, in the absence of a higher authority or arbitrator).

A model for collaborative problem-solving in dialogue based on the notion of negotiation has been described by Baker (1994). The model was originally developed for person-machine educational interactions (Baker, 1989) ; subsequently it was developed to model collaborative problem-solving, having been validated with respect to several dialogue corpora for several different tasks in the domain of physics (Baker, 1995).

Although we can not discuss this model in detail here, the basic idea is that collaborative problem-solving proceeds by a negotiation process, defined as a type of interaction where the agents have the mutual goal of achieving agreement with respect to an as yet unspecified set of *negotia*, under certain constraints (relating to the problem, the social situation, the knowledge states of each agent, …). Such a final state may be achieved by three possible strategies : *mutual refinement* (each agent makes proposals, each of which are transformed by the other), *stand pat* (one agent only makes proposals, with different forms of feedback, encouragement, discouragement, …, from the other) and *argumentation* (conflict in proposals is made explicit and mutually recognised, each tries to persuade the other to accept their proposals). Although knowledge may in fact be more or less indirectly co-constructed during each strategy (e.g. during 'constructive argumentation' — see Baker, 1996), here we shall concentrate on the most frequent used and typical strategy: *mutual refinement.*

Each strategy is defined in terms of a set of *communicative acts* and sets of *relations* (created by dialogue operators) that are established between the propositions that they express. The basic communicative acts for the mutual refinement strategy are OFFER and ACCEPTANCE or REJECTION. These are defined using Bunt's (1989) model for dialogue. OFFER's have the following most important pertinence condition (when uttered by agent A1) : "accept(A2,p) → accept(A1,p)".

In other words, OFFERs are *conditional* communicative acts, that can be interpreted as follows : A1 will accept the proposition p (a problem solution, an action …) *iff* A2 will do so ("I will if you will"). Acceptances and rejections have the function of allowing the agent that made the original offer to accept its own offer or not (on the basis that the other does so).

For our purposes here, this view of communicative action in collaborative problem-solving has important implications. It means that the information expressed by collaborating students should not be viewed as transmitted knowledge that can be acquired by their partners (re the 'information flow' view mentioned in introduction), but rather as the expression of more or less tentative proposals, 'to be refined', that will be retained in the common unfolding solution if mutually accepted.

OFFERs and ACCEPTANCE/REJECTIONs rarely occur in isolation, but rather in sequences, and the sequential position of communicative acts produce additional secondary effects on the contexts of agents. For example, if A1 offers "We are in France", then A2 offers "we are in Lyon", the second offer indirectly communicates acceptance of the first, in virtue of the informational (logico-semantic) *relations* between the contents of the two offers ("Lyon is in France" & "we are in Lyon" → "we are in France"). Similarly, "We are in Lyon", followed by "We are in France" could, in certain contexts, communicate *rejection* (i.e. we are in France, but I don't agree that we are in Lyon), via a Gricean implicature (see later discussion). This is why it is also important to study the relations between communicative acts in this strategy, that — at least on the *knowledge* level — may be defined in terms of dialogue operators, or *transformation functions*.

Transformation functions (TFs) are described in terms of the *logico-semantic relations* that are established between the propositions expressed in pairs of communicative acts, either of the same speaker or between speakers. The two communicative acts do not have to directly follow each other in the dialogue. The claim that relations exist between propositions expressed in communicative acts is of course a simplification — but one that most often works — since a given proposed proposition relates in fact to *the previous context*, from the agents' own point of view. Thus, a given utterance will sometimes relate to what is implicit in a previous one, or what it is taken to imply. This point will be taken up in discussion.

The following subsections summarise the basic transformation functions ("TFs"), with examples taken from a specific corpus of collaborative problem-solving dialogues. The students' task was to find a way of representing the rebound behaviour of balls of different substances (from the experimenter's point of view, the aim was that the students would make some progress towards discovering the coefficient of restitution).

The basic model underlying the sets of TFs is as follows. When two agents, A1 and A2, are engaged in collaborative problem-solving, when one agent makes a specific proposal with respect to the problem's solution, the second can respond to it in basically four ways :

- by *expanding* the proposal (elaborating it further, generalising it, …)

- by *contracting* the proposal (making it more specific, reducing its informational content, …)

- by providing *foundations* for it (justifications, explanations), or

- by remaining *neutral* with respect to its informational content (either verbatim repetition, or some type of reformulation of the way it is expressed in language, conceptualised, …).

3.1 Expansion transformation functions ("TF"s)

These transform the initially offered proposition by extending or elaborating it in some way, either its 'scope' along a generality/specificity dimension, or in informational terms (adding a proposition, inferring a new one on the basis of the initial offer). The following are typical example TFs in this class.

Expansion TFs		
TF name	Transformation	Examples
Generalisation	p =TF=> generalisation of p	" … the energy ..." =TF=> "... all the energy ..."
Conjunction	p =TF=> $p \wedge q$	"it's nil on arrival ..." =TF=> "it's nil at the start, and it's nil on arrival .."
Specific-value	A(?_x) =TF=> ?_x = φ (A - predicate; ?_x - variable)	"...kinetic energy " =TF=> "is theoretically nil!"
Inference	p =TF=> $p \rightarrow q$	"potential energy will increase" =TF=> "therefore the rebound force increases"

Table 2: Set of expansion transformation functions (Baker, 1994)

Often, the transformation takes place in a way which is left partially *implicit* . For example, if one student offers p1, then the second may apply the conjunction TF simply by offering p2 ; if it is not mutually believed that p2 is contradictory with p1, then the second offer may be interpreted as "OFFER($p2 \wedge p1$)".

3.2 Contraction TFs

These functions are usually the inverse of expansion functions : they restrict the previously offered proposition, or render it more specific (less general). For example, an OFFER "the ball moves at 10 centimetres per second" that follows "the ball has slowed down" renders the latter more specific. However, contractions are not always the inverse of expansions. For example, the inverse of inferring a new proposition q from a previous proposition p (p =TF=> $p \rightarrow q$) could be seen as proposing that q implies p (p =TF=> $q \rightarrow p$), in other words, giving a reason or justification for p. This case therefore comes into the *foundational* class (see below). Empirically, the transformation that most often occurs with respect to inferences is to deny their validity.

Contraction TFs		
TF name	Transformation	Example
Conjunction-contraction	$p \wedge q$ =TF=> p	"the force and the speed … " =TF=> "… the force …"

Exclusive-disjunction-choice	$p \vee q$ =TF=> p	"the mass or the density" =TF=> "rather the density"
Contra-inference	$p \rightarrow q$ =TF=> $p \wedge \neg q$	"since it rebounds lower that shows it's the friction" =TF=> "it's lower, but it's not the friction that's involved !"
Subtype	p =TF=> sub-type p	"... all the energy, ..." =TF=>"... the kinetic energy ..."
Class-restriction	$p \in \{S\}$ =TF=> $p \in \{\alpha\}, \alpha \supset \{S\}$	" … do you really think kilogramme ball made of rubber would rebound a lot ?" =TF=> "yes, but only in the case of a perfectly elastic rebound"

Table 3: Set of contraction transformation functions (Baker, 1994)

The contraction strategy can be viewed as knowledge *deconstruction*. Although theoretically, inverses of all expansion functions could exist, in reality examples are hard to find. One possible explanation is in terms of cognitive economy : one the students have co-constructed a possible solution by expansion, if they recognise "that's not it, it can't be that !", it may be easier to simply let the solution drop and start again (perhaps taking part of the previous solution) rather than to deconstruct (contract) it piece-by-piece.

3.3 Foundational TFs

These provide foundations (reasons for/against, explanations) for offered propositions. Often, this occurs at the end of an expansion phase, when the students 'step back' and attempt to verify or check the current joint proposal. For example, in Figure 1:

“it’s nil on arrival, in fact..." ==> "... since the object stops"

Counter-reasons can indicate a shift to the argumentation strategy, although more isolated occurrences may occur within the mutual refinement strategy when weighing the 'pros and cons' of a proposal that is in fact mutually agreed.

3.4 Neutral TFs

These leave the content of the initially offered proposition either completely unchanged or else transform its meaning, expression in language or conceptualisation. They often operate at the level of negotiating *understanding* and *agreement, rather than at the knowledge level itself.* This is why exact *repetitions* (nil transformation at knowledge level) usually have the function of confirming understanding and agreement. For example,

"... the object stops, ...” =TF=> "... it doesn’t move, ...".

Often this occurs as a 'termination' to an expansion phase, when the students try to summaries the current joint solution. Some transformations on the level of language (or terminology) are very important from a learning point of view. For example, in

"with the weight" =TF=> "… the mass"

the students pass from everyday language, and conceptualisations ("weight") to the target (scientific) language/conceptions ("mass").

4. Applying Michalski's operators for analysing a dialogue

To evaluate the interoperability of learning and dialogue operators, we attempted to apply Michalski's operators to a short dialogue extract[2] (figure 1). This illustrates the way in which knowledge is transformed (negotiated) in dialogue using the mutual refinement strategy. Necessarily, the extract only illustrates some of the knowledge transformations or transmutations that could occur. In this short dialogue, the task involves students conducting experiments where they try to discover the properties of balls of different substances that could explain their rebound behaviour — in fact, the coefficient of restitution.

	(…)
[1] A1:	So what can we say if there’s an inelastic impact ?
[2] A2:	Well, that the energy .... all the energy ....
[3] A1:	Well, that the kinetic energy is theoretically nil !
[4] A2 :	It's nil on arrival, in fact …
[5] A1:	Since ... since the object stops, in fact, oh yes, especially since there it doesn’t move, uh...
[6] A2:	it's nil at the start and nil on arrival … about energy … yes, but at the moment of an inelastic impact, what is it that …
[7] A1:	we've been doing that for a while now ! <sighs>
[8] A2:	but we've also …
[9] A1:	wait, … inelastic impact, right, you've conservation of momentum, but … the kinetic energy isn't conserved ! I think that's what we've seen … with elastic impact, by contrast, both are conserved
[10] A2:	Yes, elastic impact, there's the total energy which is conserved …
[11] A1:	Yes
	(…)

Figure 1. Example of mutual refinement strategy in physics collaborative problem-solving dialogue.

Figure 2 shows a graphical analysis of the extract in Figure 1, using Michalski's knowledge transmutations. The different propositions expressed by each agent (learner) are analysed in separate columns; relations between them, according to the Michalski operators, are shown as labels on arrows.

General observations. The extract begins by A1 proposing the focus of the discussion: "inelastic impact" ([1]). This is accepted by A2, and remains the common focus until the second part of [9], when it shifts to the case of "elastic impact". In this first discussion on the case of inelastic impacts, the joint solution is successively transformed basically by successions of *generalization* and *specialization* of what is to be discussed within the "inelastic impact" focus - energy, kinetic energy - and by adding more details to the reference set (class of

[2] An analysis of the same extract, using transformation functions, is presented in Baker (1994, 1995).

inelastic impacts) by *concretion* - kinetic energy is nil on arrival at the ground, energy but not momentum is conserved. Once this solution element has been jointly transformed as far as the agents can or want to, A1 then moves on to consider the inverse case — elastic impact.

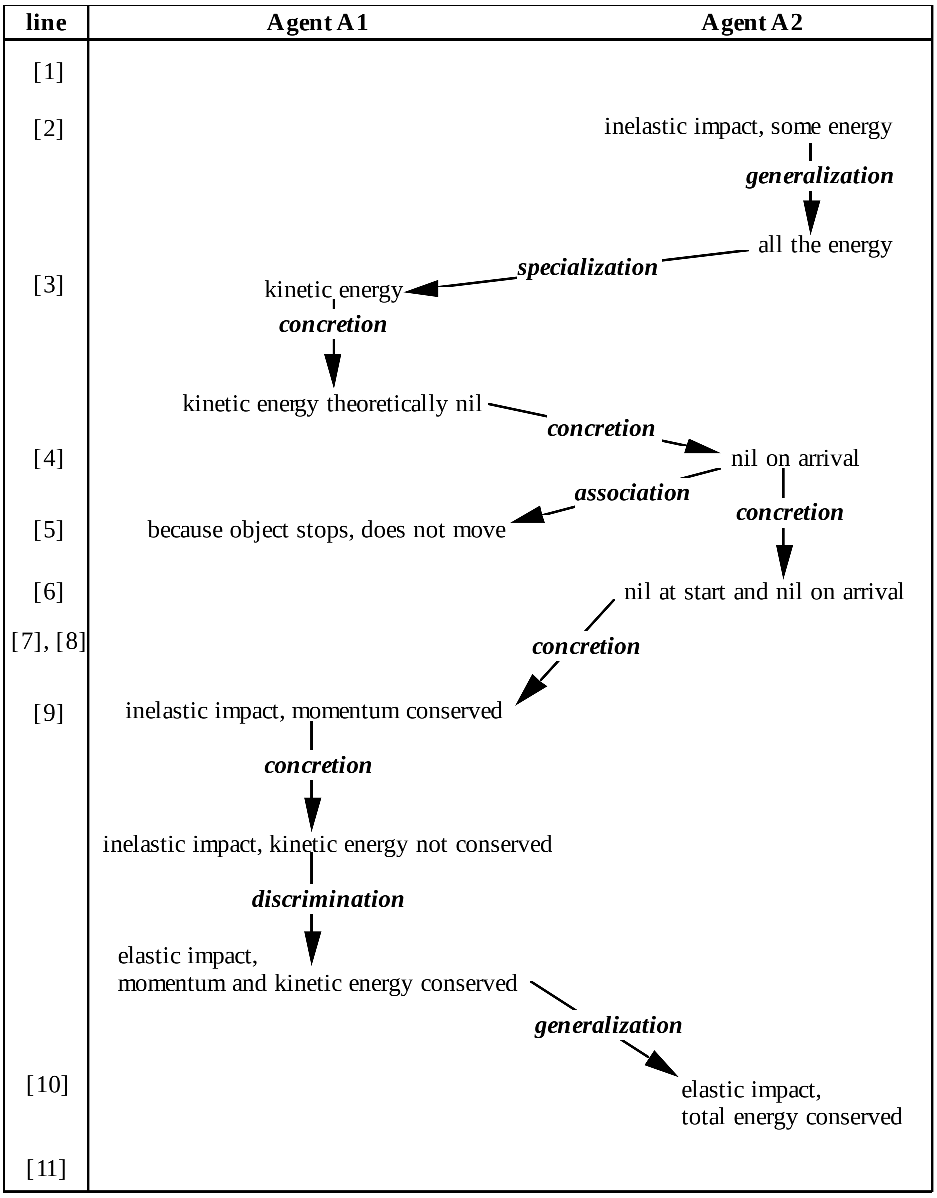


Figure 2. Analysis of example physics collaborative problem-solving dialogue using some Michalski's machine-learning operators.

Transformations and collaboration. The expressed propositions and transformations are distributed across both agents. An important aspect is 'who transforms what, expressed by whom?'. Thus, agents may transform propositions that they themselves have expressed, either within their turn or across turns. More importantly - for the study of collaborative learning - they transform propositions expressed by others. Such a graphical analysis thus provides one way of mapping out the extent to which agents are really 'working together', i.e. collaborating (Baker, 1995). Roschelle and Teasley (1995) have described some similar phenomena as "collaborative produced production rules", distributed across collaborative completions of utterances (the first utterance provides the antecedent of a rule, the second its consequent). When agents are not collaborating (resolving in parallel), this can be seen from the fact that transformations are usually performed by agents on their own contributions across turns.

What is the degree of goodness of fit ? In each transmutation analysed, the most appropriate operator has been chosen. Nevertheless, certain aspects are left out of the analysis, even when restricting consideration purely to the knowledge level, and more fine-grained distinctions can be made. Thus, within line [5], although agent A1 appears, on the surface, to give a reason for why the kinetic energy is nil on arrival at the ground, this does not really transform knowledge itself into new propositions ("because the object stops, it does not move"). Rather, the sequence "nil on arrival"-"object stops"-"does not move" is a sequence of *reformulations* of the meaning of "kinetic energy nil", in more everyday language. A second difficulty with applying the operators can be seen from line [9], where A1 concludes discussion of "inelastic impact" and begins to consider the case of "elastic impact". Is this transformation really discrimination (apparently the best fit), i.e. " determines a description that discriminates the given set of entities from another set of entities" ? It is analysed as discrimination here since, by stating that energy is conserved with an elastic impact, this discriminates this case from that of an inelastic impact where energy is *not* conserved. However, in the case considered here, what is important is that the agent moves from considering a class of entities "impacts that are inelastic" to considering the negation of it, i.e. "impacts that are not inelastic (= elastic)". This does not actually *transform* the knowledge base, but rather shifts the focus of attention so that all of the set of "impacts" will be considered. Many of these differences are taken into account by analysis in terms of transformation functions (see Baker, 1995).

What is left out of the analysis ? In addition to there being finer knowledge-level distinctions that can be made with respect to the propositions that are analysed, some utterances are of course left out entirely, since they do not express or transform new knowledge. It is interesting at this point to briefly mention what they are, for this specific example, and to describe their functions[3] :

[3] Note that parts of some utterances that are in fact analysed are also left out of consideration here — for example, the "wait ..." at the beginning of line [9], that relates to coordination of turns. Clearly, we also do not consider purely social aspects of interpersonal interaction, such as maintaining "face", friendship, and so on.

Elements left out of the analysis :

[1] A1: So what can we say if there’s an inelastic impact ?
(…)
[7] A1: we've been doing that for a while now ! <sighs>
[8] A2: but we've also …
(…)
[10] A2: Yes<…>
[11] A1: Yes

Line [1] is left out of the analysis since it does not express a new proposition, nor transform knowledge. Rather, it proposes/introduces a new sub-problem to be *focussed* on. Lines [7]-[8] express frustration with lack of progress in joint problem-solving, from the point of view of A1. In these terms, it can be viewed as part of metacognitive control of 'knowledge transformation'. Finally, lines [10] and [11] act as a 'punctuation' to the knowledge transformation sequence. They do this given that the agents must reach agreement, that a sequence of transformations is to be terminated, and that it is so due to joint agreement. All of these aspects missed out of analysis relate to a single fact about dialogue and collaborative problem solving: it needs to be *controlled* or *managed* (Bunt, 1995).

ML systems use heuristics (background knowledge, types of learned knowledge, generalisation hierarchies, etc...) to control the learning process. These heuristics are generally applied directly to the implemented transmutations functions. This control remains is purely internal since it appears in a one-shot learning process. As mentioned by Wrobel (1994), these ML systems run like a compiler and do not support the cyclic process of knowledge acquisition or problem-solving.

In summary, analysis of collaborative problem solving dialogues with a specific set of ML operators can in fact give us a picture of how collaboration has taken place, on a certain level of generality. Aspects that are specific to (human) interactions such as negotiation (reformulation) of meaning and interaction management are of course not included in such an analysis. This fact may have important implications given that the effort required to ground interaction, to create joint meaning may be important for collaborative learning (Schwartz, 1995; Baker et al, *this volume*) and that interaction management problems may hinder it.

5. Applying dialogue operators to analysing a machine-learning algorithm

In the previous section we applied machine learning operators to a dialogue. We now perform the inverse, applying the set of dialogue operators to a specific inductive learning algorithm, developed by Mitchell, Utgoff and Banerji (1984). In this case, a concept is built from a set of positive and negative instances. The 'version space' is the set of all encountered instances. This space is represented by storing the most specific and the most general definitions of the concept to be learned. Here we do not enter into the details of the algorithms using this approach, but the principle can be summarised as follows: generalise the most specific description (hereafter S) such that it includes all positive instances, and specialise the most general definition (hereafter G) such that it excludes all negative instances.

5.1 Analysis of example

In order to articulate this approach with the set of dialogue operators discussed previously, table 4 presents the case of two artificial agents who conduct an instance-based argumentation (using instances as the main source of evidence) in order to develop their own hypotheses and mutually criticise them. In this fictitious dialogue, Agent 1 has a bias towards over-specialisation, maintaining S and criticising G. Conversely, Agent 2 is biased towards over-generalisation, maintaining G and criticising S. This example is based on a rudimentary description language - the version spaces S and G are defined by the conjunctive set of (attribute object value) triplets- and a simple generalisation hierarchy. Tautological predicates (e.g. colour object (white or grey)) are removed from descriptions.

Table 4 illustrates the convergence of S and G towards a set of hypotheses (S8 and G7) which match the set of considered instances. Any agent can criticise the hypothesis of its partner (generalising S as in step 2 or specialising G as in step 5), but can also update its own hypothesis as in step 3.

The dialogue is simple because it is assumed that the descrption language and the generalisation hierarchy is shared by the two agents. In a real collaborative learning from examples, those aspects should themselves be grounded in dialogue (see Baker, Hansen, Joiner & Traum, *this volume*). For instance, the generalisation at step 2 can be either to substitute the class 'square' by the larger class 'rectangle'. An alternative would be use an exclusive disjunction operator and produce "Either a square on the bottom side or a rectangle on the upper side". The dialogue would be richer if the agents had different background knowledge (description language, generalisation hierarchy). Agents should then negotiate between two possible generalisations (e.g. at step 2, between a larger class or the disjunction of smaller classes) or even negotiate a predicate (e.g. Agent-1 asks "What do you mean by 'equilibrated' ? " and Agent-2 answers "Instances 1, 2 and 3 are 'equilibrated'"). The hypotheses S and G would more likely converge if the agents were provided with heuristics which gave priority to attributes which have been negotiated with their partners. Richer interactions would also occur if each agent incompletely ran its algorithm (not exploring all possible generalisation/specialisations), hence generating criticisable hypotheses (e.g. over-general hypothesis in step 4) or providing partial criticisims (as in step 2).

What the example shows is that the overlapping between Michalski’s learning operators and Baker’s dialogue operators is not suprising given that one can view an inductive approach as the negotiation of the lower and upper limits of the version space.

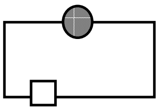
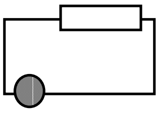
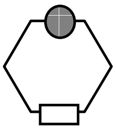
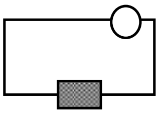
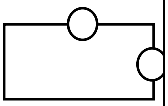
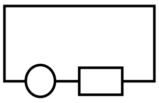
	Instances	**Agent**	**Fictitious dialogue** (with dialogue transformation functions)	**Evolution of S**	**Evolution of G**
1	Positive 	A1	*"A Spuc is a rectangle with a small white square on the bottom side and a grey disk on the upper side"* (generates S1)	(shape O1 rectangle) (shape O2 square) (colour O2 white) (size 02) < (size O1) (shape O3 disk) (colour O3 grey) (position O2 (bottom-side O1)) (position O3 (up-side 01))	
2	Positive 	A2	*"Actually, your white square can be a rectangle"* (applies generalisation TF)	(shape O1 rectangle) (shape O2 ~~square~~ rectangle) (colour O2 white) (shape O3 disk) (colour O3 grey) (position O2 (bottom-side O1)) (position O3 (up-side 01))	
3		A1	*"Ok then but the rectangle can be on any side"* (applies generalisation TF)	(shape O1 rectangle) (shape O2 rectangle) (colour O2 white) (shape O3 disk) (colour O3 grey) ~~(position O2 (bottom-side O1))~~ ~~(position O3 (up-side 01))~~	
4		A2	*"I would just say it is 3 polygons"* (applies generalisation and conjunction-contraction)		(shape O1 polygon) (shape O2 polygon) (shape O3 polygon)
5	Negative 	A1	*"Not simply 3 polygons: the large polygon must be a rectangle otherwise this instance would be positive."* (applies counter-reason)		(shape O1 ~~polygon~~ rectangle) (shape O2 polygon) (shape O3 polygon) (size 02/O3) < (size O1)
6	Positive 	A2	"Right but the two small ones can have any colour: here the small rectangle is grey and there the disk is white" (applies generalisation)	(shape O1 rectangle) (shape O2 rectangle) ~~(colour O2 white)~~ (shape O3 disk) ~~(colour O3 grey)~~	
7	Négative 	A1	*"Ok, but you see that the two small polygons must be a disk and a rectangle otherwise this instancs would be positive"* (applies subtype)		(shape O1 rectangle) (shape O2 ~~polygon~~ rectangle) (shape O3 ~~polygon~~ disk) (size 02) < (size O1)
8	Negative 	A1	*"Moreover they cannot be on the same side otherwise this instance would positive".* (applies specific-value)	(shape O1 rectangle) (shape O2 rectangle) (shape O3 disk) (position O2 (side-X O1)) (position O3 (side-Y 01)) side-X side-Y	

Table 4. Describing an inductive approach to machine learning as a dialogue. (~~Value or Predicate~~ = modifed in concept definition, Value or Predicate = added to concept definition)

5.2 Comments on the example

The above analysis shows how simple concept learning can take place through a dialogue between two artificial agents. This occurs through an interplay of three processes expanding the definition of the concept, contracting it, and by invalidating over-generalisations with reference to new instances of the concept.

The process begins with three successive generalisations. In line 2, A2 generalises the description of the square to be "any rectangle" ; in line 3, A1 proposes that the rectangle does not necessarily have to be on the upper side, but can be on any side. In fact, these two generalisations of the description are not quite of the same order. In the first case a single entity (square) is described in terms of the superclass (rectangle) ; in the second, a determinate description (top side) is universally quantified (the set of all sides in the figure). In line 4, A2 generalises "rectangle" to "polygon", and "contracts" the information about the specific colours of the shapes.

In line 5, A1 invalidates this over-general description, citing the negative instance that shows that the polygon description is too general. In line 6 A2 accepts this invalidation, and goes on to make a specific generalisation about the colour of the smaller objects that had previous corresponded simply to contracting information.

Consideration of the two negative instances in lines 7 and 8 enables the agents to produce a more specific description — the smaller polygons must be a disc and a rectangle (subtypes of class "polygon"), and they must be on different sides.

It is useful to compare this example with the dialogue analysed in figure 3, where the students were attempting to interpret results of an experiment (two balls of different substances had different rebound behavious when released above the ground). These two tasks are similar to the exent that the world outside the dialogue provides a set of phenomena that can validate or invalidate their present descriptions of those phenomena/instances. Thus specific experiments with the rebounding balls correspond to specific presentations of instances. However, the main difference lies in the fact that the students are not told specifically whether the experiments confirm or invalidate their descriptions (interpretations). Often this is not clear-cut, and is at the origin of much of their dialogue.

6. Theoretical comparison

The analyses presented in the previous sections show machine learning operators can be applied for describing knowledge refinement in collaborative problem-solving dialogues, and that machine learning can be presented dialogically. However, there are a number of important differences that need to be considered in our comparison. They concern the number of agents, the use of context in communication, the necessity to control dialogue and the relations between agents' mental states and their utterances.

6.1 The number of agents

An obvious difference between dialogue operators and ML operators is that the former relate contents of communicative acts uttered by *different* agents, whilst the latter relate knowledge states of the *same* agent. Nevertheless — at the knowledge level at least — this difference is not taken into consideration. Firstly, dialogue operators do function very well as monologue operators as in lines [2] of figure 2 (A2 generalises his own statement) or in lines [3] (A1 concretises his own statement). Conversely, single-agent learning operators can be viewed as dialogue operators in multi-agent learning : an agent A, using operator X to solve a conflict between two divergent goals, can be re-implemented into a multi-agent systems, in which two (sub-)agents A1 and A2 have respectively each of these goals and negotiate with the same operator X.

Dialogue and learning operators can be adapted for the case of, respectively two or one agent, because they are intrinsically binary operators, i.e. operators which describe the differences between two knowledge states (as stored in a system or expressed by agents), without indication as to whether these knowledge entities belong to one or more agents. The distinction between one or more agent is in any case not intractable since some distributed cognition theories define a group as a single agent (Salomon, 1992) or, conversely, some scholars view the individual as a group of agents (Minsky, 1987). As pointed by Dillenbourg (*this volume*), terming something an agent is a theoretical or methodological choice, given that different levels of granularity reveal different aspects of cognitive processes.

6.2 Context and communication

Whilst in a ML system, the application of a specific transmutation conveys a unambiguous knowledge change, this is not the case when knowledge transformations are expressed as utterances in dialogues between human agents. In fact, in natural dialogues, the apparent, or 'surface', knowledge transformation that takes place may not directly correspond to the indirect transformation that is communicated and mutually understood. This is because speakers use the mutually understood context as a means for allowing their hearers to make *inferences*, that will in turn correspond to the real knowledge transformation that takes place.

As is well-established in pragmatics, the context within which 'inferences to meaning' are made must contain general 'cooperative principles' (Grice, 1975), as well as knowledge relating to the domain of discourse. Meanings that are conveyed via such principles are termed "implicatures".

Consider the following pair of utterances :

[1] A1 : It is a whale
[2] A2 : It is a mammal

On the surface level, a *generalisation* transformation has been applied here : the object under discussion, initially described as a whale is now described as a mammal. There is nothing contradictory between being a mammal and being a whale, since a whale is a mammal. However, in certain dialogue contexts [2] could in fact be interpreted as a *rejection* of [1]. How does this work ? Let us assume that A1 and A2 mutually know that a whale is a mammal, and that they both assume the cooperative maxim of "informativeness" ("make your contribution as informative as required"). In this case, [2] appears to be a violation of this maxim, since it states what is already commonly assumed. So A1 infers that A2 must be intending to communicate something else, i.e. that the object is a mammal, but not (necessarily) a whale. So the communicated knowledge transformation (invalidation) is not the same as the apparent one (generalisation).

The principle reason why speakers use such indirect means of communication is that this enables greater economy of effort in speech — the context can be assumed and exploited without having to be rendered explicit on every occasion. What does this imply for the possible fusion of dialogue and ML operators ? It means that a ML system that incorporates person-machine interaction based on a unique set of operators will be faced with the following choice: either, allow implicit language-use for the human user, which would be economical for users but difficult for a machine to understand since it impleis to maintain a representation of the context, or else impose a completely explicit dialogue that would be easier for the machine but tiresome for the user. No doubt an appropriate balance between the two could be found.

6.3 The relation between knowledge states and utterances.

We mentioned previously that the knowledge transformation indirectly communicated in dialogue does not necessarily correspond to the transformation that would be proposed from an analysis of the surface contents of successive utterances. A similar problem resides in the fact that the contents of speakers' utterances do not necessarily have direct relations to their mental states.

In general, agents can be viewed as believing what they say, according to a basic assumption of sincerity. However, there is no infallible algorithm for inferring mental states from utterances (although there may be non-monotonic 'default' methods - e.g. Perrault, 1990). For example, if one student says *" … it's the different densities that explain the rebounds*", what does this mean ? That the student *believes* that proposition ? Not necessarily. In fact, students often make "proposals" — "might it be x ?" — without necessary commitment to believing them (Baker, 1994), especially when they are attempting to solve a problem that is beyond their present ability.

Neither do agents necessarily transform the contents of immediately previous utterances. As mentioned above, the agents knowledge spaces are almost never rendered entirely explicit in communication, simply because this would make communication highly uneconomical. Hence, a given utterance often does not relate *directly* to a previous one, but rather to the agents perception of its underlying context. In the example below, A2 responds in fact to the following proposition attributed to A1 (what A1 is perceived to be implying/implicating):

"the steel ball rebounded higher than the rubber one, so the differences in mass explain the different rebound behaviour".

[1] A1 : look, the steel ball rebounded higher than the rubber one
[2] A2 : it's not a question of mass !

Finally, agents do not of course necessarily believe what they hear. There is a probability that the receiver (1) does not hear correctly, (2) does not understand, (3) understands but disagrees, (4) understands and agrees but cannot integrate in his knowledge base, ... There is no guarantee that agents directly and completely *internalise* (in a Vygotskian sense; (see Littleton & Hakkinen, *this volume*)) the knowledge publicly constructed in dialogue. In this case, if the receiver of an offer accepts it, then the original speaker can infer that both (s)he him/herself and the listener can now accept it (see the pertinence condition above). Thus, offers (if accepted) can lead to mutual acceptance.

6.4 Dialogue control and controlling the learning process

A further gap that appears between dialogue and learning concerns *control*, of both problem-solving and dialogue (see previous §4). Whilst Michalski's taxonomy does not include 'management' operators, control is however critical in many learning processes. For instance, in incremental learning from examples, the selection of relevant examples and counter-examples directly influences the learning outcomes. The issue is to know if the operators used in dialogue management and in learning management are similar. The interoperability of learning and dialogue operators should hence distinctively discussed at two levels: the *knowledge level* and the *control level*.

The existence of implicit meanings and indirect uses of communication indicate that collaborative problem-solving in real human dialogues can not be simply viewed as a direct expression of knowledge transformations. Transformations take place within a dialogue context that is assumed to be mutually understood, but which is rarely completely internalised. Surface transformations in dialogue do not correspond directly to what is *learned*.

These points concern the differences between knowledges states and utterances, rather than the related operators. However, the deep linguistic and philosophical questions raised by the knowledge/utterances discrimination inevitably come into play in the comparison of learning and dialogue operators. However, in the remainder of this chapter, we attempt to see to what extent these theoretical problems can be bypassed for practical goals.

7. Practical implications

7.1 Multi-agent learning

Despite these fundamental issues concerning the gap between mental states and utterances, can the (partial) interoperability of dialogue and learning operators contribute to implement learning in multi-agent systems (Weiss & Dillenbourg., this volume)? We examine this issue, with our own perspective where modelling human collaboration prevails on engineering issues.

Unfortunately, dialogue and learning have traditionally been studied in different branches of artificial intelligence (AI). Traditionally, AI has been clustered into disciplines such as problem solving, ML, computational linguistic, robotics, vision, tutoring systems, … . Agent architectures reflect this history: they generally dissociate some core reasoning layer (knowledge representation, problem solving, learning,...) from the interface layer (incoming and outgoing messages, vision, ...). There have been a few attempts to merge different techniques and efforts to unify AI sectors e.g. around SOAR (Rosenbloom & al, 1993), but globally, the independent development of the different AI dispciplines does facilitate synthetic approaches such as looking for what is common between learning and dialogue models.

The first step to develop an algorithm which unifies learning and dialogue is to improve the interoperability of learning/dialogue operators. People Power (Dillenbourg & Self, 1992) illustrates this principle. The artificial agent uses the same operators for reasoning and for dialoguing: agree or refute. Dialogue was based on a binary tree of arguments where any argument could be agreed or refuted, where refutations could then on their turn be agreed or refuted, and so on. When the artificial agent reasoned alone, it used the same operators with itself, i.e. it was able to refute its own arguments, its own refutations, and so forth. In other words, reasoning was implemented as an inner dialogue. Learning resulted from the fact that the artificial agent replayed - *modus modendi* - some parts of previous dialogue during its inner monologues.

In the task, the two agents have to modify an electoral systems in order to gain more seats for their party. When ‘proving’ (in the EBL sense) that his action will will to gain seats, agent-1 (using over-general rules) argues that agent-2 that : 'If a party gets fewer votes, then it will get fewer seats'. Agent-2 attemps to prove that this statement is wrong in the particular case at hand and proves it: " Since this party has no seats, it cannot loose seats". Agent-1 will store this refutation, and check it the next time it will trigger the over-general rule. This relation between the over-general rule and its refutation is functionnaly equivalent to specializing the over-general rule by adding has a condition the negation of the refuting rule: 'If a party gets fewer votes and has seats then it will lose seats'.

The principle of similarity between dialogue and reasoning was applied here in its simplest way, the set of operators being extremely simple (agree/disagree). Real dialogues are of course more complex. An avenue for research is to design similar dialogue/learning mechanisms but with a richer set of dialogue/learning operators and communicative acts, or systems which are able to negotiate 'below the rule level', i.e. not by using conflictual rules, but by negotiating during the rule instanciation process (Dillenbourg, 1996).

The notion of "reasoning as a dialogue with oneself" illustrates the applicability of dialogue operators as learning operators. The reverse applicability, using learning operators to describe group interactions, can be implemented if one sees collaborative dialogues as the process of building a shared knowledge set. Individual utterances can hence be seen as learning operators with transform the shared knowledge set.

However, the shared knowledge space is not a publicly accessible entity, as in the case of knowledge states in a ML system - representations of it exist in each agent. Some discrepancies are actually important for learning, provided that they are detected and resolved. This brings us back to a point mentioned earlier concerning the multidimensionality of negotiation. In addition to domain-level relations, established by transformations, agents also transform the meaning of utterances, in order to check what was meant, to establish mutual understanding of the joint space - a process that has been described as "social grounding" (Clark & Schaefer, 1989). Such grounding or negotiation of meaning is required even in the case where agents attempted to make all of their relevant knowledge states publicly accessible.

In other words, there is something "stereoscopic" in collaboration : Agent A exploits differences between his own knowledge and his representation of Agent-B's partner knowledge. This inter-partner comparison is different from what we saw so far. Learning operators describe differences between successive knowledge states, dialogue operators between more or less successive utterances. In both cases, this sequentiality limits somewhat the scope of potential differences. At the opposite, the differential reasoning on two independent knowledge states must cover any possible difference, until an empty intersection. The set of operators used both in learning and dialogue should hence be extended to cover the mutual modelling process which supports the construction of shared knowledge.

It is likely that the perspective of ‘multi-agent learning as a dialogue among agents’ - which would be a new perspective that has been ignored so far - opens new and promising research directions (Weiss & Dillenbourg, *this volume*).

7.2 Integrating user interaction in machine learning

ML systems most often induce a general concept from a number of examples by generalising the common features of the examples. Among the particular difficulties in such systems are the description language limitations and the incrementality of the learning process. Valid algorithms may produce incorrect results from correct data, simply because there is some mismatch between the way in which data is provided and the way it is processed. This is especially important as we deal with imperfect domain theories, where initial data may be incomplete or may contain many biases. Knowing this, users can modify data that they provide to the system in order to allow it to refine the knowledge that it has already built.

To reduce such difficulties, different ML systems or models have been proposed, and their common purpose is to move from the one-shot learning process, and take into account interaction with expert-users. There is a growing interest in integrating interactions with expert users in ML systems. The goal is nevertheless not strictly to implement collaborative learning systems. The main motivation is to gain robustness by adding user-system interactions. Of course, most ML systems allow very restricted types of interactions with the user. These interactions are generally implemented by incorporating the ML system into two kinds of environment:

- Interface environment: In such environment, the systems that have been exhibited tend to place more emphasis on interaction than on learning. These systems belong to the area of 'programming by demonstration (PDB)' (Cypher, 1993). The learning systems are used to analyse user behaviour and then try to improve the user's task (see for example, Yoshida and Motoda, 1996).

- Knowledge acquisition environment: In such environment, the systems focus more on the learning/acquisition process than on interaction. Knowledge acquisition is viewed as a cyclic process that allows the systems to refine its learned or acquired knowledge. Examples of such environment are the knowledge based system MOBAL (Morik & al., 1993), and the learning environment LEGAL (Mephu Nguifo, 1997).

LEGAL has its foundations from a learning model reported by (Mephu Nguifo, 1994, 1995). In such model, the learning system interacts with an expert-user in order to: (1) learn knowledge from initial data provided by the expert, (2) derive new results from learned knowledge, and (3) also help the expert-user during the interpretation of learning results. LEGAL receives initial input (binary data and learning parameters) from the user. The user is considered to be an expert of the domain. The learned knowledge arises from the successive application of different transmutation functions such as specialization, selection, characterization, generalization and discrimination over the lattice structure of the binary data.

The derivation of new results is a way to validate the learned knowledge. Unseen data are provided by the user to test the efficiency of the learned knowledge. LEGAL uses different kinds of derivation transmutations for this purpose. It can either use deductive derivation or analogical inference by combining reformulation and similization transmutation functions.

As the user works on an incomplete model, the last step becomes very important since both expert and system can change their knowledge state depend on the results interpretation. To allow this, the learning model includes the notion of proofs and refutations (Lakatos, 1984; Hayes-Roth, 1986) through the mechanisms of objections. Objections are built by the system as a way of proving its results. An objection is true until the user refutes it as an explanation of the system decision. Whilst the acceptance of objection could change the user knowledge, the refutation of an objection should allow to modify the knowledge learned by the system.

This system is based on a static and cyclic process for controlling learning. It has been extended to a dynamic model of control (Mephu Nguifo, 1997). This extended model adds an indirect dialogue between the user and the system, by integrating various tools with different purposes, and which are linked together in order to dynamically control the learning process.

Nevertheless, these systems basically remain cooperative (rather than "collaborative", in the sense described here - see also the introduction, Dillenbourg, *this volume*) since there is a fixed division of labour between the system and the user: the former has to learn, explain, or recognize, and the latter must choose the initial data and analyse the system's results. We are seeking more collaborative systems where the user and the system can *negotiate* the data, the amount of noise, the heuristics, ... In collaboration, the system and the human share roughly the same set of actions (Dillenbourg & Baker, 1996). Hence, any reasoning step performed by the ML system has to be available also to the user through some dialogue function. The interoperability of dialogue and learning operators may improve the collaboration between a human agent and the system agent in two ways: (1) by increasing the modularity of functions, each agent being able to contribute of a smaller step of the learning process, (2) by increasing the symmetry of the distribution of functions, most functions being allocable either to the human or to the machine agent. Another potential advantage of incorporating dialogue and learning together is that this might lead to ‘self-explanation effects’ or the ‘effect of understanding by explaining’.

8. Conclusions

This chapter compared two sets of operators which come from different research communities. Learning operators have been proposed in ML where knowledge states are directly inspectable. Dialogue operators come from psychology and linguistics, they describe relations between the content conveyed by the subjects actions, namely by their utterances, via a commonly assumed context. The gap between dialogue and learning operators reflects the gap between knowledge and action. For instance, we pointed out that dialogue operators cover the 'strategy' aspects of dialogue (e.g. saying something one does not believe in order to check one's partner agreement), while Michalski's learning operators do not cover these 'strategy' aspects (e.g. selecting which example to consider next). While the interoperability of learning and dialogue operators seems feasible at the knowledge level, it seems more difficult to achieve it at the strategical level.

Finally, we mentioned that learning operators could be applied to the analysis collaborative dialogues, not by modelling individual knowledge states, but by tracing the emergence of a body of shared knowledge. Our recent experiments on computer-supported collaborative work (Dillenbourg & Traum, 1997) show that this shared knowledge is — to a large extent — reified on a whiteboard. A whiteboard is a drawing software in which two users can edit the same page together, modifying or deleting each other’s objects. The information displayed on the whiteboard remains there and is mutually visible, the subjects use it as the main medium to co-construct the solution. Namely, they select for display on this persistent medium the information which is itself persistent, i.e. will remain valid along the solution process. Hence, the shared knowledge, this invisible result of grounding dialogues, is reified by a single set of observables notes, which brings this linguistic concept closer to the machine learning framework.

9. References

Baker, M.J. (1989). *Negotiated Tutoring : An Approach to Interaction in Intelligent Tutoring Systems.* PhD thesis (unpublished), Institute of Educational Technology, The Open University, Milton Keynes, Great Britain.

Baker, M.J. (1994). A Model of Negotiation in Teaching-Learning Dialogues. In *The Journal of Artificial Intelligence in Education,* **5**(2), 199-254.

Baker, M.J. (1995). Negotiation in Collaborative Problem-Solving Dialogues. In Beun, R.-J., Baker, M.J., & Reiner, M. (Eds.) *Dialogue and Instruction*, (pp. 39-55). Berlin : Springer-Verlag.

Bunt, H. C. (1989). Information dialogues as communicative action in relation to partner modelling and information processing. In Taylor, M. M, Néel F, & Bouwhuis D. G. (Eds) *The structure of multimodal dialogue,* 47-74. North-Holland: Elsevier Sciences Publishers.

Bunt, H. C. (1995). Dialogue Control Functions and Interaction Design. In Beun, R. J., Baker, M.J, & Reiner M. (Eds.), *Dialogue and Instruction, Modeling Interaction in Intelligent Tutoring Systems*. Proceedings of the NATO Advanced Research Workshop on Natural Dialogue and Interactive Student Modeling, 197-214. Berlin, Germany: Springer.

Cypher, A. (1993). *Watch what I do: programming by demonstration*. Cambridge, MA: MIT Press.

Dillenbourg, P. (1996) Some technical implications of distributed cognition on the design of interactive learning environments. *Journal of Artificial Intelligence in Education,* 7 (2), 161-179.

Dillenbourg, P., & Self, J. A. (1992). A computational approach to socially distributed cognition. *European Journal of Psychology of Education*, 3(4):353-372.

Dillenbourg, P. & Baker, M (1996) Negotiation spaces in Human-Computer Collaborative Learning. Proceedings of the *International Conference on Cooperative Systems* (COOP'96), Juan-Les-Pins (France), June 12-14 1996

Dillenbourg, P. & Traum, D. (1997) The role of a whiteboard in a distributed cognitive system. *Swiss workshop on collaborative and distributed systems*. Lausanne, May 1st 1997.

Goldberg, D.E. (1989). *Genetic algorithms in search, optimization, and machine learning,* Reading, MA: Addison-Wesley.

Grice, H.P. (1975). Logic and conversation. In Cole, P. & Morgan, J. (eds.), *Syntax and Semantics 3: Speech Acts,* 41-58.

Hayes-Roth, F. (1986). Using Proofs and Refutations to Learn from Experience. In Mitchell, T. M., & al. (Eds.) *Machine Learning: An A.I. Approach,* San Mateo, CA, 221-240.

Hobbs, J. R. (1982). Towards and Understanding of Coherence in Discourse. In *Strategies for Natural Language Processing* Lehnert, W. G. & Ringle, M. H. (Eds.) , 223-244. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Lakatos, I. (1984). *Preuves et Refutations*. Paris: Editions Hermann.

Mann, W. C., & Thompson, S. A. (1988). Rhetorical Structure Theory: Toward a functional theory of text organisation. *Text* 8(3), 243-281.

Mephu Nguifo, E. (1994). Galois Lattice: a Framework for Concept Learning. Design, Evaluation and Refinement. In Proceedings of 6th Intl. *Conf. on Tools with Artificial Intelligence*, 461-467, New Orleans (LA), November 6-9 th, IEEE Press.

Mephu Nguifo, E. (1995) Multi-Level Concept Modelling for SBL. In Proceedings of MLNet (ECML) *workshop. on Knowledge Level and Machine Learning*, Heraklion (Crete - Greece), April 28-29th, Fensel, D., & al. (Eds).

Mephu Nguifo, E. (1997) An interactive environment for dynamic control of machine learning systems. In Smith, M. J., Salvendy, G., & Koubek, R. J. (Ed) *Design of Computing Systems: Social and Ergonomic Considerations*, 21B, 31-34. Amsterdam: Elsevier Science. Proceedings of 7th Intl. Conf. on Human Computer Interaction, San Franscisco (CA), August 24-29th.

Michalski, R. S. (1993). Inferential Theory of Learning as a Conceptual Basis for Multistrategy Learning. *Machine Learning* 11(2-3): 111-151.

Minsky, M (1987) *The society of mind*. London: William Heinemann Ltd.

Mitchell, T.M., Utgoff, P.E & Banerji , R. (1983) Learning by Experimentation: acquiring and refining problem solving heuristics. In Michalski R.S., Carbonell J.G and Mitchell T.M (Eds) *Machine Learning: An artificial intelligence approach* (Vol. I, pp. 163-190). New York: Springer-Verlag.

Morik, K., Wrobel, S., Kietz, J. U., & Emde, W. (1993) *Knowledge acquisition and machine learning - Theory, Methods and Applications.* London: Academic Press.

Perrault, R. (1990). An application of default logic to speech act theory. In Cohen, P. R., Morgan, J., & Pollack, M. E. (Eds.) *Intentions in Communication*, Cambridge, MA: MIT Press.

Roschelle, J. & Teasley, S. (1995). The construction of shared knowledge in collaborative problem solving. In O’Malley, C. E. (Ed.) *Computer-supported collaborative learning*, Heidelberg: Springer-Verlag.

Rosenbloom, Laird, & Newell, A. (1993) *The Soar Papers: Readings on Integrated Intelligence*, MIT Press.

Salomon., G. (1993) (Ed). *Distributed cognitions. Psychological and educational considerations* (pp. 111-138) Cambridge, USA: Cambridge University Press.

Sanders, T. J. M., Spooren, W. P. M., & Noordman, L. G. M. (1992). Towards a Taxonomy of Coherence Relations. *Discourse Processes*, 15, 1-35.

Schwartz, D. L. (1995). The emergence of abstract dyad representations in dyad problem solving. *The Journal of the Learning Sciences*, 4(3), 321-354.

Yoshida, K., & Motoda, H. (1996). Automated User Modelling for Intelligent Interface. *Intl. Journal of Human-Computer Interaction,* 8(3), 237-258.

Wrobel, S. (1994) *Concept Formation and Knowledge Revision*, Dordrecht: Kluwer academic publishers.