Conceptual Framework for Interactive Ontology Building

Jean Sallantin, Christopher Dartnell, Jacques Divol, Patrice Duroux

A Conceptual Framework for Interactive Ontology Building

Jean Sallantin, Christopher Dartnell, Jacques Divol and Patrice Duroux LIRMM
UMR 5506 – Université Montpellier II/CNRS
161 rue Ada, 34372 Montpellier cedex 05, France
Email: js,dartnell,divol,duroux@lirmm.fr

***Abstract*— An ontology is a formal language adequately representing the knowledge used for reasoning in a specific environment. When contradictions arise and make ontologies inadequate, revision is currently a very difficult and time consuming task. We suggest the design of rational agents to assist scientists in ontology building through the removal of contradictions.**

These machines, in line with Angluin's "learning from different teachers" paradigm, learn to manage applications in place of users. Rational agents have some interesting cognitive faculties: a kind of identity, consciousness of their behaviour, dialectical control of logical contradictions in a learned theory respecting a given ontology and aptitude to propose ontology revision.

In the paper, we present an experimental scientific game Eleusis+Nobel as a framework outlining this new approach, i.e., automated assistance to scientific discovery. We show that rational agents are generic enough to support the ontology building process in many other contexts.

I. Introduction

Layered architecture and formal languages used to develop scientific, administrative, commercial or transport applications are current topics of active discussion. In this layered architecture, applications in specific domains are built with different levels of norms. Norms and standards such as OWL, SOAP, CORBA are continuously proposed, used and... contested.

With each and all of these standards, the production and particularly the revision of an ontology is quite time-consuming for an expert. Besides, for users, processes underlying the behaviour of the application are very often considered as unpredictable and unexplainable black boxes.

Our objective is to generate a *Rational agent* that learns how to manage the applications on behalf of the users. So the rational agent participates, in interaction with the experts, to the design of an ontology used to supervise applications.

Dana Angluin's theory about "learning from different teachers" [1] gives a new formal basis to interactive learning. In this theory, a *Learner* is an agent that does have the proper command of a black box of applications and a *Teacher* is another agent able to perform an interesting task from the learner's point of view: the teacher knows how to combine applications to perform a new successful application. The *learning from teachers* process leads a learner, after querying the teacher a finite number of times, toward acquiring a behaviour that simulates that of the teacher.

We use the fundamental theorem resulting from "learning from different teachers" to establish that a *Rational agent* emerges from a stable interaction cycle between a teacher and a learner. This result also provides a mathematical foundation for interactive ontology building in the light of a specific interaction between a learner and teachers.

This article is mainly dedicated to the definition of a conceptual framework in terms of constituent interactions within a rational agent and interactions between scientists and rational agents.

Content of the paper

In contemporary science, a *scientific theory* is a paradigm that is formulated by means of association of natural and formal languages, and which predictions and explanations about real world phenomena are accepted by a scientific community.

A *scientific community* publishes scientific theories and shares an experimental environment that allows testing a theory and distinguishing two theories.

In a first part: *E+N: a Scientific Discovery game*, we introduce these notions by describing an experimental environment that simulates the process of scientific discovery.

In the second part: *Learning from different teachers*, we present Dana Angluin's paradigm that reinforces the mathematical foundation of our model.

In the third part: *Contradiction based dialectic control*, we introduce a dialectic control of contradictions in paraconsistent logic that extends the framework previously defined for the learner with an online learning [2] mechanism allowing the selection of the most informative query among all possible ones for theory improvement.

In the fourth part: *Rational Agent's formal foundation*, we propose a paradigm associating formal semantic with ontology building in interaction with rational agents.

In the fifth part: *Rational Agents' supervision by scientists*, we discuss which principles can guide scientific activity with the assistance of rational agents.

We then finish by presenting the *concrete implementation of the Framework*.

II. E+N: a scientific discovery game

In order to test and improve the design of a platform where rational agents assist scientists to build an ontology interactively, we have developed a software system managing interactions within the context of the game E+N (for Eleusis + Nobel).

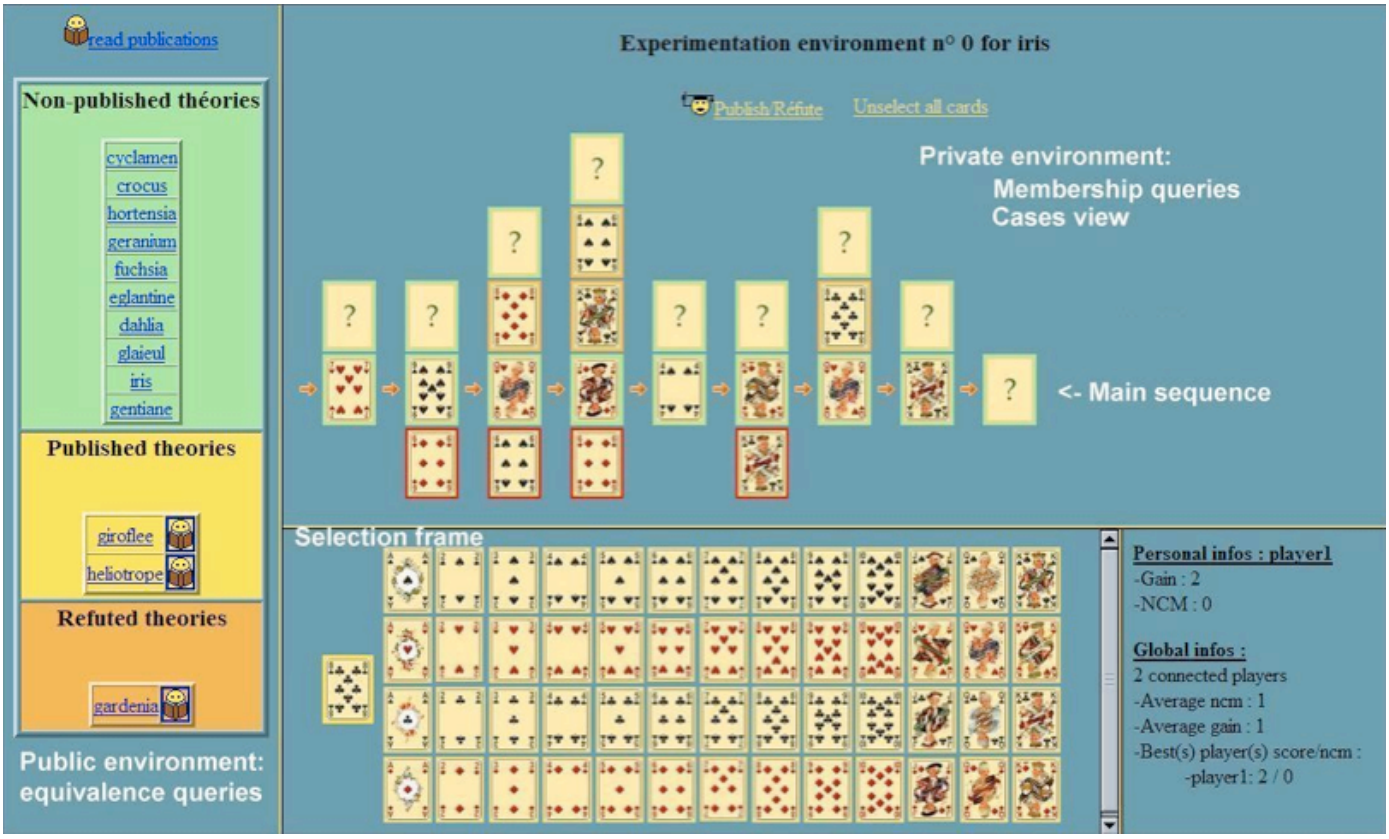


Fig. 1. Eleusis + Nobel Game display

This game is inspired from Abbot's famous Eleusis card game [3]. In this game, the goal of the player is to discover a hidden rule (for example, red/black cards alternation) that simulates a "universal nature's law", and determine all the possible cards' sequences that can be played. The problem's complexity is determined by the length of the sequence needed to determine the next card, and by the fact that the rule can be deterministic or non-deterministic. It is then possible to change the rule to adjust the complexity of the problem and to fix the difficulty level of learning problems. However, for any cards' sequence, the hidden rule must allow to determine at least one following card to ensure that the rule describes a continuous process.

For each player, the results of his/her ongoing experiments are always visible in his/her private workspace. However each player cannot see other players unless they are members of a scientific group. That will be outlined later.

In the following, we illustrate step by step how a player operates on this platform.

1) Each *player* chooses a rule he wants to study, from a publicly accessible set of hidden rules in the left-hand side of the screen. Rules are accessible by an imaginary name but their meaning is hidden and the player can switch between hidden rules whenever he wants to.
2) Consider in Fig.1 the depicted Main sequence, that consists of the set of eight cards plus a "? hole". A *player* selects one out of fifty two cards and decides where to put it on the Main sequence. There are two possibilities: either the position chosen is the next available one on the right hand side (RHS), or it is a position in the Main sequence already occupied by a card. In the first case the player tries to extend the Main sequence. In the second case, he tries to modify the value of one of the already existing position of the main sequence. In this case, he is forced to put his/her card on the top of the stack of cards belonging to the position. This choice ends the activity of a player in a single move.
3) The *"Nature" evaluates* if this new sequence is compliant with the hidden rule and displays the result of its evaluation:
   - The card is put on the RHS to extend the Main sequence.
     - If the card is acceptable, then it's surrounded in green.
     - If not, it is surrounded in red and put under the rightmost "? hole" of the Main sequence.
   - If the card is proposed to substitute another one in any other position, there are three possibilities.
     - Either it is an acceptable substitution for the card in the main sequence at the selected position, then it is marked by green, and a new question mark is made available on top of the position.
     - Either it is not acceptable, then the card is queued on the bottom of the position, under the Main sequence's row.

– Finally, if it is compatible with the Main sequence's cards in previous positions but not in following positions, then it is surrounded in orange.

After analyzing these results,

1) A *player formulates* a theory on the basis of these results.
   The theory can be expressed formally or semi-formally, for example in natural language. This theory intends to approximate or to coincide with the hidden rule. We call *ontology* the terms and relations among terms used by the player to describe the experiment. These terms are used to formulate the theory.
2) Therefore the *player may publish a new theory* explaining the nature's law he is studying. In this case, the player publishes his/her ontology and the experimental data justifying his theory.
3) The *player can also refute a theory* published by another player, in which case he produces a counter-example.
4) Players are all members of a *scientific community* that gives or suppresses credits when they publish or refute. Such events are immediately communicated to all players.
5) In this game, a *rational agent is the assistant* of a scientist or of a scientific group. The scientist formulates an ontology in order to describe the cards and the cards' sequence.
   The Rational Agent assists a player by:
   - Indicating its own predictions,
   - Designing experiments,
   - Formulating a theory,
   - Anticipating refutations on a publication draft.

This collective behavior is regulated by rules that we call *interaction protocol*. This interaction protocol between players is inspired by the Nobel game created by David Chavalarias [4] to reproduce collective research situations. This game was modeled to gather information on human behaviors in scientific research situations, under various conditions. It is based on a Popperian conception of scientific research: the activity of scientists belonging to a scientific community consists in formulating hypothesis and refuting them. We have used this protocol to manage *equivalence queries* described hereafter. We have also extended the facilities in the Eleusis game to manage *membership queries*. Even if the Nobel game was originally concerned with human players, we extended it to our rational agents in order to validate the interaction cycle between the user and the rational agent presented later.

Here are enumerated E+N's major concepts that will be used in our research:

1) E+N gives the same spatio-temporal referential to every player in the scientific community.
2) An experimentation is a *membership query*. By using this type of query, each scientist asks the Nature if the sequence x respects the hidden rule f.
3) Each Scientist's interpretation of a hidden law is then biased by the sequences produced by his previous experimentations.
4) A publication is an *equivalence query*. By using this type of query, each scientist asks the other scientists to confirm or to refute his theory g (can you prove that my theory is false on the basis of your experiments?) . A publication contains an ontology, a theory and an experimental sequence confirming the theory, and possibly references to other publications that are re-used in the theory.
5) Every scientist can use a published sequence as a basis to prove or refute a published theory.
6) The *principle of reducibility* is verified in this framework: every formal demonstration is always reduced to the visual form of an experimentation.
7) The *principle of nominalization* is also verified: every relevant regularity used in the formulation of the theory (for instance "red/black alternation") identifies a process that can be re-used to detect this regularity in sequences.
8) In this game, each player gives his/her own meaning to cards. This meaning depends on what rules he believes the card must respect to be present in the sequence. In other words, each card changes the state of the "environment", and influences the next player's actions.

The first E+N experiments presented the following empirical properties:

1) Human players have interesting emotions when they play, especially when they are refuted or when they are the first to publish.
2) When 20 rules are hidden, each of them constraining sequences of two cards, the time required for a scientific community of 10 players to publish a stable set of publications is between one and two hours.

Finally, scientific advances come from "a perpetual revision of contents by improvement and erasure" [5]. In E+N, a dialog drives a game in which each player tries to win by leading the other ones to admit their contradictions, by publishing refutations of their theories. This game assumes that "the generative necessity in Science is not an activity but a dialectic".

III. Learning from different teachers

In this section, we present Angluin's "Learning from different Teachers" paradigm [6]. In this formalism, a rational agent is the result of a stable interaction process between a teacher and a learner. We first present cognitive considerations, then formal results. Finally, we justify the importance of giving a formal foundation to the constructive definition of a rational agent.

A. Cognitive considerations

Dana Angluin's formalism is based on the following cognitive considerations. All humans are physically similar, but they are singular from a cognitive point of view. In other words, they don't know exactly what are their functionalities: individual functionalities are for each human a non-dominated black box. Nevertheless, even if we do not perform in the same way, we still know how to learn from each other by imitation

since we have the similar aptitudes. Practically, we are able to learn how to juggle without understanding the teacher, without neither the time for introspection nor the capacity to perform a theory of juggling.

To summarize these considerations:

- Both the teacher and the learner have the same global cognitive architecture.
- Both of them are able to perform the same universal tasks, but using personal strategies and tactics.
- The teacher is able to solve the problem.
- Whatever might be the learner's questions, the teacher is able to answer them.
- The imitation process stops when the learner succeeds in simulating the teacher's functionalities.
- Imitation does not require the agent to reason about his intentions, believes or desires.

B. Formal results

Concerning the previous cognitive considerations, Dana Angluin argues that:

- An agent is the combination of an operating system and an applicative black box.
- To solve a problem, an agent uses his operating system to combine applications.
- Both the learner and the teacher are such agents.
- The learner doesn't dominate the process of selecting an adapted sequence of applications to solve the problem.
- The learner can query the teacher.
- The teacher knows how to solve the problem by combining applications.
- The teacher is able to answer any query.

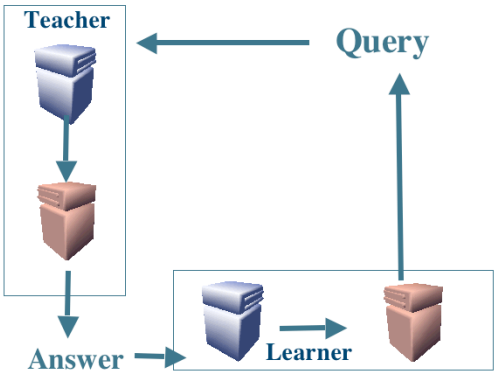


Fig. 2. Angluin's Learning protocol

In order to study the convergence of the learning process, Dana Angluin turns the problem into a theoretical problem.

The formalism is the following:

- The hidden functionality to be learned is a recursively enumerable function $f : N \rightarrow N$.
- The applicative black box is a black box of recursively enumerable functions.
- The operating system is a recursively enumerable function combining the applications.
- Every teacher using its operating system and its black box is able to compute the hidden functionality.
- The computational performances of the learner's black box are comparable to the computational performances of the teacher's black box.
- For any x belonging to N, a teacher is able to answer the "membership query Me.Q." $f(x) = y?$.
- For any function $g : N \rightarrow N$ proposed by the learner, a teacher is able either to answer the "equivalence query" $f = g$, and to provide x such that $f(x) \neq g(x)$ when the answer is negative.

This formalization allows demonstrating the following theorem: *Whatever its applicative black box might be, and whoever its teachers are, a learner proposes after a finite number of queries a solution producing only a finite number of errors if:*

1) *The learner is instructed by teachers who have already solved the problem.*
2) *The computing performances of the teachers are comparable to the learner's ones.*

Let us comment this result:

1) The teaching process never stops but converges towards a stable cycle.
2) The system (learner-teacher) is singularized by the applications of the learner's black box, i.e.: each learner computes each universal function differently.
3) The learner is permanently able to interact with new teachers.

Dana Angluin completes these results by technical results coming from her previous work:

1) This theorem remains true even if the teacher is malicious -*it will mislead the learner a finite number of times*- and cautious -*it will prefer to stay quiet a finite number of times*-. [6]
2) The convergence process of this "learning from different teachers" is similar to Gold's "language identification in the limit". It is also a special case of Valiant's PAC learning [7].
3) The "equivalence" and "membership" queries enable to learn logical theories. Their number is used to estimate the convergence of the learning process [8].
4) The duration's measure of the learning process is called a *dimension*. Different dimensions are given by the maximal number of queries required by the learner to produce, suggest or eliminate an hypothesis [9].

C. Why Angluin's theory is important

Angluin's theory is a paradigm using natural and formal languages to explain Learning. In this specific case, the natural language is restricted to the following terms: black box, application, agent, teacher, learner, query, and operating system. The formal language mainly defines these terms with the notion of recursive function, and queries are formalized by logical statements. In the following, we first show by means of experimentations in Law, how this theory makes sense in a realistic interactive ontology building. Then we discuss the formal relevance of Angluin's theory.

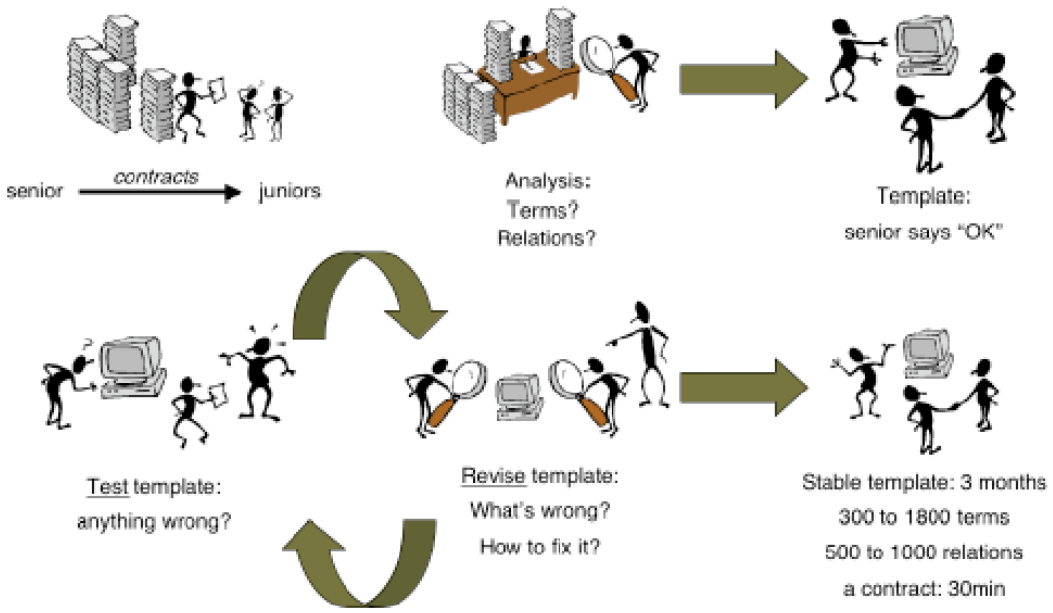


Fig. 3. Ontology building in Law

1) Empirical relevance: In this section, we present a practical application of Angluin's theory in Law [10].

In this experimentation, the Teacher is a senior lawyer who has a very high competence in a specific legal domain: negotiating contracts. The Learner is constituted by a couple of junior lawyers who have no specific competence in this domain. A single constraint propagator stands as an applicative black box to assist them in formulating ontologies and expressing queries to the expert. The expert gives them a set of contracts, then the juniors extract relevant terms and relations between terms in order to propose a first theory f to the senior. If the senior agrees with the theory, they learn from Teacher's examples a more specific theory g. Then they design a new contract x that matches the Teacher's theory f but doesn't match their theory g. So they formulate the membership query M.Q. $f(x)$ to the senior. The senior tells them how to modify the theory f in order to eliminate this counter-example x, and the juniors repeat the operation. The end of the process is reached when the senior estimates that the contract x produced by the juniors is exotic but not irrelevant. In this application, the protocol is an "online learning" since the juniors choose the membership query Me.Q. $f(x)?$ in order to create the most informing example about $f(x) \neq g(x)$, giving them a way to revise their theory.

In this application, there is no symbiotic relation between the teacher and the juniors because the teacher is not able to access the juniors' application. The query is formulated by a contract that is written by the juniors in order to control the bias of their learning set. The result of this application is to produce a theory about a category of contracts. This concrete application shows how to improve Angluin's theory by introducing the notions of ontology and theory to formulate queries.

2) Formal relevance: Angluin's formalism gives such a strong basis to the interactive learning from different teachers that we consider it as a foundation for our conceptual framework. To do this, we must detail the notions of operating system and application to facilitate the use in this formalism, of notions such as Ontology, Theory, and Contradiction management that have been showed essential by the previous experiment.

For instance, let us consider the notion of Ontology. By *Ontology*, we intend terms and relations between terms allowing the formulation of a relevant description of real world objects. Practically, this notion is used to formulate logical constraints on the application's input and output in such a way that the *operating system* is able to manage the application's launching and halting rules. Therefore, it is imperative to precisely define the notions of operating system and applications to allow an ontological control.

This shall be done in section V.

D. Rational Agent software requirement

By *rational agent software*, we intend a software, consisting of the combination of one or more Teachers (operating system + applicative black box) and a Learner (operating system + applicative black box) respecting the conditions given by the theorem to ensure a robust learning that enables the agent to reach a stable cycle after a finite number of interactions.

This rational agent software shows two formal advantages:

- The Rational agent software is a network of processes which are continuously regenerated during interactions.
- The stable invariant of a rational agent software is its own organization, as it will be discussed later.

The rational agent software presents the following cognitive aspects:

- A rational agent has a subjectivity since it is able to learn universal laws by its proper way.
- The interaction cycle gives the rational agent a consciousness of its behavior.
- A rational agent has no purpose, belief or intention, but it looks like having them for the observers (Dennett's stance).
- In order to teach and use a rational agent, a human acts on the rational agent's Teacher component.

In order to estimate the performances of our own rational agent, it seems useful for us to make a parallel with the major steps in a child's cognitive evolution:

- A purely Angluin-like rational agent can be compared to a four-months-old baby who is merely able to build his identity during the interactions with his mother.
- A rational agent using dialectical control of contradictions to select queries, as shown in next section, would be an eight-months-old child who can merely say "no" to his mother.
- A rational agent able to analyze the world by doing experimentations and communicating his assumptions to others would then be seen as a child who explores his world and expresses what he understands, as a little scientist.

IV. Dialectical control of Contradictions

As we said in the introduction, an ontology is not a one person's subjective design of terms and relations between terms but a consensual choice made to manage applications. Ontologies and theories are formulated in a paraconsistent logic [11] in order to allow the learner to be aware of his theoretical errors in such a way that the learner's contradictions are the interaction triggers.

In this section, we suppose that the operating system and the applicative black box are controlled by logical rules. These rules come from an ontology produced by the teacher. Thanks to this ontology and using the teacher's examples the learner is able to produce a predictive theory. A dialectical control takes place when the learner is able to analyze contradictions coming from the differences between its theory and the teacher's ontology in order to select the most informative queries.

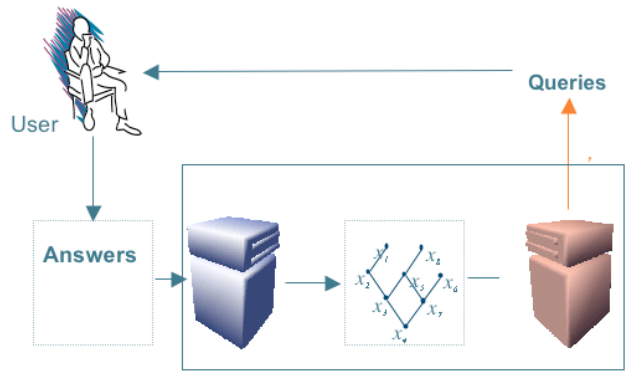


Fig. 4. Human acting on behalf of a teacher in a Rational Agent

In this section, we respect the previous process illustrated in the Law application:

1) The teacher formulates the ontology of a domain.
2) The learner formulates its own theory from examples described using the teacher's ontology.
3) A dialog between the teacher and the learner is used to analyze contradictions between the learner theory and the teacher ontology.
4) When errors are detected on an example, the learner uses them to correct its theory.
5) When the error comes from the problem's formulation, the teacher corrects his ontology.

The following section introduces paraconsistent logics to manage contradictions.

A. Contradictions in paraconsistent logic

Experimentation is a way to reveal contradictions in a scientific theory. In logic, a contradiction is a statement that is simultaneously true and false. In a classical logic, a theory cannot present a contradiction without being trivial since all the statements are theorems if a contradiction exists. On the contrary, paraconsistent logic [12] allows non trivial theories in presence of contradictions.

In a language L having a negation $\neg$,

- A theory T is contradictory if two theorems A and $\neg A$ belong to this theory.
- A theory is not trivial only if all the formulae in L are not theorems.

In a paraconsistent logic, two formula A and $\neg A$ can simultaneously be true.

The following deduction rule gives the classical contradiction's principle.

$$\frac{\neg A \vdash B \quad \neg A \vdash \neg B}{A}$$

$$\frac{\neg \text{"}pope\text{"} \vdash \text{"}rain\text{"} \quad \neg \text{"}pope\text{"} \vdash \neg \text{"}rain\text{"}}{\text{"}pope\text{"}}$$

The following deduction shows how the production of a contradiction in a paraconsistent logic requires four arguments.

$$\frac{\neg A \vdash B \quad \neg A \vdash \neg B \quad \neg A \vdash \neg(B \wedge \neg B)}{A \wedge \neg A}$$

$$\frac{\neg pope \vdash rain \quad \neg pope \vdash \neg rain \quad \neg pope \vdash \neg(rain \wedge \neg rain)}{pope \wedge \neg > pope}$$

In this example, all the arguments are evaluated. Only if the contradiction $\vdash \neg(rain \wedge \neg rain)$ is not admitted, "it is contradictory to be pope".

The Da Costa's paraconsistant logic is monotonous and only four out of the sixteen Morgan's Law are verified:

$\neg(\neg A \vee \neg B) \vdash (A \wedge B)$
$\neg(A \vee B) \vdash (\neg A \wedge \neg B)$
$\neg(\neg A \vee B) \vdash (A \wedge \neg B)$
$\neg(A \vee \neg B) \vdash (\neg A \wedge B)$

The following section explains the use of these Morgan's laws by the learner for building a theory.

B. Learner's theory formation in a paraconsistent logic

In this section, we present on a toy problem the abductive process that is used by a learner to produce a theory.

In this toy problem, the teacher's examples are numbers. The numbers are described by the following attributes $small, medium, strong, verystrong$. The attribute value is $+$ or $-$ whether the attribute is verified or not. So they are called positive and negative attributes. The code is such that two successive numbers differ by only one attribute.

- $0+ : small- \wedge medium- \wedge strong- \wedge verystrong-$
- $1+ : small+ \wedge medium- \wedge strong- \wedge verystrong-$
- $2+ : small+ \wedge medium+ \wedge strong- \wedge verystrong-$
- $3+ : small+ \wedge medium+ \wedge strong+ \wedge verystrong-$
- $4+ : small+ \wedge medium+ \wedge strong+ \wedge verystrong+$
- $5+ : small- \wedge medium+ \wedge strong+ \wedge verystrong+$
- $6+ : small- \wedge medium- \wedge strong+ \wedge verystrong+$
- $7+ : small- \wedge medium- \wedge strong- \wedge verystrong+$

Let us define the notions that are required to manage learning.

- By *Prototype*, we intend a conjunction of positive and negative attributes that formulates a membership query Me.Q..
- By *Example*, we intend an instance of prototype given by the teacher to the learner.
- By *canonical form*, we intend a clause that formulates an equivalence query.
- By *Teacher's Ontology*, we intend a network of canonical forms with the prototypes verifying them.
- By *Learner's Theory*, we intend a network of canonical forms and prototype verifying them learned from Teacher's examples.
- By *Abduction*, we intend the process of learning a theory.

There is no formal difference between the Teacher's ontology and the Learner's theory. But the Learner's theory building depends on the Teacher's ontology. In the illustration, the initial teacher's ontology has an empty set of canonical forms and prototypes that are the numbers. Let us explain how the teacher refines the ontology by taking into account the learned theory. We suppose that the Learner's learning set is given by the even numbers 0+, 2+, 4+, 6+.

The learning method used here is a Galois' lattice learning method that extracts a specific set of regularities given by a set of irreducible elements of the lattice [13].

- Equivalence queries:
  - Does "medium+ $\Longleftrightarrow$ small+ " ?
  - Does "medium- $\Longleftrightarrow$ small- " ?
  - Does "verystrong+ $\Longleftrightarrow$ strong+ " ?
  - Does "verystrong- $\Longleftrightarrow$ strong- " ?

These regularities show what is the learning bias: in the learning set, the equivalence query "Does medium+ $\Longleftrightarrow$ small+?" expresses that on this data, "medium+" is equivalent to "small+".

The figure 5 shows that the objective is to learn a theory under useful contradictions from the teacher's point of view.

The abduction method consists here in considering that the regularities are prototypes verifying a canonical form. Then, the Morgan's laws are used to find the clauses which formulate the canonical forms implying the prototypes.

1) $\neg(\neg medium \vee \neg small) \vdash (medium \wedge small)$
2) $\neg(medium \vee small) \vdash (\neg medium \wedge \neg small)$
3) $\neg(\neg verystrong \vee \neg strong) \vdash (verystrong \wedge strong)$

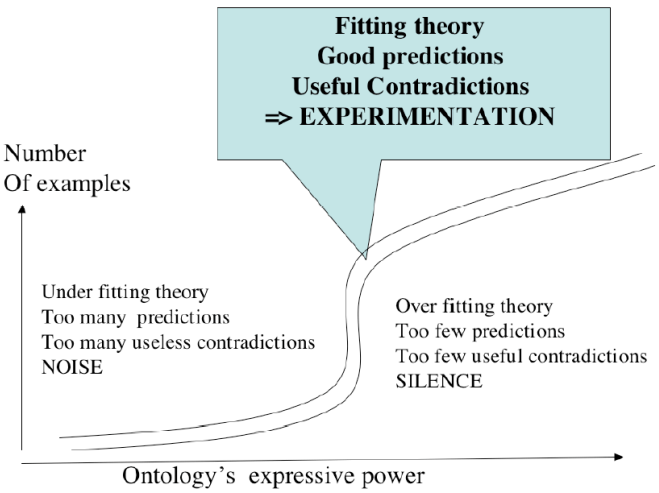


Fig. 5. Relation between contradiction, ontology expressivity and experimentation's requirement

4) $\neg(verystrong \vee strong) \vdash (\neg verystrong \wedge \neg strong)$

The abduction based on Morgan's laws uses a negation, which may produce contradictions when applied on new cases.

Remark: the $\Longrightarrow$ relation translates a causal assymetric link. The symbols + and - denote modalities about terms. By combining them, we obtain some specific metarules that define what kind of regularities can be transformed into clauses. For instance, suppose that the membership query Me.Q. does not mix positive and negative, existential and contingent modalities. The following observed regularity,

- "This +" $\Longrightarrow$ "that + "

may be interpreted as:

- "This exists" $\Longrightarrow$ "that exists"
- "This appears" $\Longrightarrow$ "that appears"

And the following observed regularities,

- "This -" $\Longrightarrow$ "that - "
- "This does not exists" $\Longrightarrow$ "that does not exists"
- "This disappears" $\Longrightarrow$ "that disappears"

These links express a membership query Me.Q.. given by a logical clause

- Is it true that "son exists $\Longrightarrow$ Father exists" ?
- Is it true that "father not exists $\Longrightarrow$ "Son not exists"?
- Is it true that " if smoke appears" $\Longrightarrow$ fire appears" ?
- Is it true that " fire disappears" $\Longrightarrow$ "smoke disappears"?

C. Contradiction's dialectical analysis

Paraconsistent logic allows to reason in presence of contradictions. Since *reductio ad absurdum* is not allowed, the only way to overcome contradictions is dialog. Let us consider that the teacher's and the learner's theories are computed in a paraconsistent logic.

In Angluin's theory, the dialog's speech acts are the equivalence and membership queries. In our model, when some contradictions occur, a dialog is required to overcome them.

1) The dialog between the teacher and the learner respects the teacher's domination and is always achieved by a teacher's "victory".

2) During the dialog, the learner is able to increase the similarity between its theory and the teacher's theory.
3) The teacher is also able to change its theory, resetting at the same time the learner's one.
4) The dialog uses membership and equivalence queries.

In the previous section, we presented how teachers and learners produce a theory with canonical forms and prototypes. We now present a way of negotiating an agreement with the help of membership queries using canonical forms and prototypes. In order to link canonical forms and prototypes to membership queries, the query $f(x)$? must be rewritten into two questions "is x correct in term of prototype?" and "is this prototype correctly linked to a canonical form ?". If the answer is yes, $f(x)$ means that x is a prototype respecting f.

This Agreement process is the one illustrated by the lawyers' application: Juniors propose their contract prototype to the senior and he estimates a correction in terms of his canonical forms.

The following example illustrates the contradiction's dialectical analysis. The analysis of a contradiction is the analysis of the contradiction's effect during an equivalence query. Let us denote $even$ the learner's theory. The Learner evaluates this theory on 7.

Using 4, we deduce :

$$\frac{7 \vdash \neg verystrong \quad \neg verystrong \vdash \neg strong}{7 \vdash \neg strong}$$

Using a paraconsistant logic's inference rules, we infer that there is a contradiction about 7 if we suppose

$\vdash \neg(strong \wedge \neg strong)$

and we locate the regularities which originated the contradiction.

$$\frac{7 \vdash strong \quad 7 \vdash \neg strong \quad 7 \quad \vdash \neg(strong \wedge \neg strong)}{7 \wedge \neg 7}$$

Observing that even(7) is contradictory, the learner asks the membership query Me.Q. f(7)? to the teacher. The teacher says ok. Then the learner adds 7 in its learning set and learns new rules.

The Learner's equivalence queries are:

- medium+ $\Longleftrightarrow$ small+
- medium- $\Longleftrightarrow$ small-

The learner's membership queries are now the following:

- 0+ $\Longrightarrow$ Verystrong- Medium- Small-
- 6+ $\Longrightarrow$ Strong- Medium- Small-
- 7+ $\Longrightarrow$ Medium- Small- Verystrong+ Strong-
- 4+ $\Longrightarrow$ Strong+ Medium+ Small+
- 2+ $\Longrightarrow$ Verystrong+ Medium+ Small+
- Strong+ $\Longrightarrow$ Verystrong+
- Verystrong- $\Longrightarrow$ Strong-

In this toy problem, the learner's theory converges towards the teacher 's ontology.

To conclude this section, we shall now present how the theory and the ontology are combined. In the first case, the ontology is given by its canonical forms and prototypes. In the second case, the ontology is given only by a complete description of the prototypes. We show the equivalence of the results in these two approaches bounding the vast set of possible combinations.

The ontology terms are the following one: nationalities (french, english, chinese), profession (spy, musician, sailor) and location (home1, home2, home3).

1) Ontology with canonical forms: The ontology expresses that a prototype must verify one and only one nationality, profession and location.

The learner's theory is:

1) $\neg(\neg French \vee \neg Chinese \vee \neg English)$
2) $\neg(\neg Home1 \vee \neg Home2 \vee \neg Home3)$
3) $\neg(\neg Musician \vee \neg Sailor \vee \neg Spy)$
4) $\neg(Home1 \vee Home2) \wedge \neg(Home1 \vee Home3) \wedge \neg(Home2 \vee Home3)$
5) $\neg(Spy \vee Sailor) \wedge \neg(Spy \vee Musician) \wedge \neg(Musician \vee Sailor)$
6) $\neg(French \vee Chinese) \wedge \neg(French \vee English) \wedge \neg(English \vee Chinese)$

The learning set is given by the following teacher's prototypes:

- $English$: "$English+$" $\wedge$ "$Home2+$"
- $Chinese$: "$Chinese+$" $\wedge$ "$Musician+$"
- Spy : "$Spy+$" $\wedge$ "$Home1+$"

Then the learner's theory is:

1) $\neg(\neg Spy \vee \neg Home1) \vdash (Spy \wedge Home1)$
2) $\neg(\neg English \vee \neg Home2) \vdash (English \wedge Home2)$
3) $\neg(\neg Musician \vee \neg Chinese) \vdash (Musician \wedge Chinese)$

Adding learner's theory to the teacher's theory gives the correct result, *i.e.* "the spy is french".

2) Ontology without canonical forms: The learning set is a complete description:

- $English : French- \wedge English+ \wedge Chinese- \wedge Home1- \wedge Home2+ \wedge Home3- \wedge Spy- \wedge Musician- \wedge Sailor+$
- $Chinese : French- \wedge English- \wedge Chinese+ \wedge Home1- \wedge Home2- \wedge Home3+ \wedge Spy- \wedge Musician+ \wedge Sailor-$
- $Spy : French+ \wedge English- \wedge Chinese- \wedge Home1+ \wedge Home2- \wedge Home3- \wedge Spy+ \wedge Musician- \wedge Sailor-$

The learned equivalence queries are:

- Spy+ $\Longleftrightarrow$ Home1+
- Spy+ $\Longleftrightarrow$ French+
- Musician+ $\Longleftrightarrow$ Home3+
- Musician+ $\Longleftrightarrow$ Chinese+
- Sailor+ $\Longleftrightarrow$ Home2+
- Sailor+ $\Longleftrightarrow$ English+
- Sailor- $\Longleftrightarrow$ Home2-
- Sailor- $\Longleftrightarrow$ English-
- Musician- $\Longleftrightarrow$ Home3-
- Musician- $\Longleftrightarrow$ Chinese-
- Spy- $\Longleftrightarrow$ Home1-
- Spy- $\Longleftrightarrow$ French-

The learned membership queries are:

- Chinese+ $\Longrightarrow$ Sailor- Home2- English- Spy- Home1- French-
- English+ $\Longrightarrow$ Musician- Spy- Home3- Home1- Chinese- French-
- Spy+ $\Longrightarrow$ Sailor- Home2- English- Musician- Home3- Chinese-

- Home1+ $\Longrightarrow$ Sailor- Home2- English- Musician- Home3- Chinese-
- French+ $\Longrightarrow$ Sailor- Home2- English- Musician- Home3- Chinese-
- Musician+ $\Longrightarrow$ Sailor- Home2- English- Spy- Home1- French-
- Home3+ $\Longrightarrow$ Sailor- Home2- English- Spy- Home1- French-
- Sailor+ $\Longrightarrow$ Musician- Spy- Home3- Home1- Chinese- French-
- Home2+ $\Longrightarrow$ Musician- Spy- Home3- Home1- Chinese- French-

The translation of the previous learned prototypes into a theory by using Morgan's laws gives a correct theory.

In this section, the Teacher's ontology and the Learner's theory are described in a paraconsistent logic. The control of the learned theory's adequacy is done via a dialectical process. We illustrated on simple examples how theories are derived from the ontology, when examples are given to the learner. The Rational agent's knowledge evolves when contradictions appear, by refinement of the ontology.

V. Rational agent's foundation

In this section, we propose a formal foundation of a rational agent which control is carried out through the use of a paraconsistent logic.

In Angluin's paradigm, the informal notions of agent, teacher, learner, operating system, and application are linked to the formal notion of recursive functions. In the previous sections, we introduced new informal notions: rational agent, theory, ontology and we linked the notion of theory and ontology to membership and equivalence queries.

In computer science, mathematical objects are used to define a formal semantic of computational objects. By this way, the consistency of computational objects is reduced to the consistency of the mathematical theory used to define these computational objects. With respect to this approach, Dana Angluin's theory defines informal notions such as "operating systems" and "applications" as being formal calculi with recursive functions. In order to preserve Angluin's results, we have to give to the new notions a formal semantic compliant with her formalism.

Many theoretical formalisms use the category's theory to give a formal semantic to the computational objects. For instance, Dana Angluin's formal semantic is a category whose objects are recursive functions and whose arrow is the composition law. Respecting this approach, we suppose that a computational category $COMP$ is given, whose objects are recursive functions and whose arrow is composition. Considering the applications, we distinguish the external actions that are used to start and to stop the applications from the internal actions that are used to manage the applications' execution.

- Therefore, the *operating system*'s operators act on the category $COMP$ with some operators that activate and halt applications.
- The *applicative black box*'s operators acts on category $COMP$ in order to perform an application.

In a rational agent, the "operating system's" operators and "applicative black box's" operators are controled using logical rules. So their operators, inherited from the logical internal operations, are used to combine canonical forms and prototypes, and to link prototypes to canonical form, as we have seen in the previous section.

To give a formal semantic to our notions, we use the category of endofunctors. . An endofunctor acts inside the category $COMP$ by operating transformations on the computational objects and on their relations. For instance, a specific endofunctor "forget functor"acts by suppressing all the relations between objects.

$f : COMP \longrightarrow COMP$.

The objects of the endofunctor category are the endofunctors. Since the endofunctor's composition is transitive, associative, and with identity, the category's arrow is the composition.

The composition of endofunctors produces diagrams. Some of them have a triangle or a square pattern. When we assign these diagrams to be commutative diagrams, we consider that the two different ways to produce objects give the same objects. On the contrary, the fact that the diagram does not commute is used to locate a contradiction between two sequences of transformations.

These violations of commutative diagrams are used to produce halting and firing actions. For instance, a rational agent's dialectical contradiction corresponds to the violation of a commutative diagram and activates the sending of a message to the user.

A. Operating system

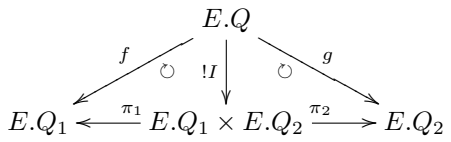
An operating system acts directly on computations in order to control the activation of applications. We define formally an operating system as being a specific category of endofunctors. We call "Equivalence query E.Q" the objects of these category. Each Equivalence query acts on a firing or halting condition.

The idea is that the operating system halts or sends a message to the user when no Membership query is a response to its Equivalence query. When an Equivalence query is discomposed in more precise Equivalence sub-queries, the operating system halts or sends a message to the user if it detects violations in the commutative diagrams built with the Equivalence sub-queries.

When the operating system is controled by logical rules, the *Equivalent queries* produced are canonical forms. As the canonical forms are combined by conjunctions and by compositions, the Equivalence queries must also be combined by abstraction of these operations that are the product and the composition.

1) Definition: By *Operating system*, we intend a monoidal category for the composition whose objects, called *Equivalence queries* are endofunctors of $COMP$. On this category, we suppose a product $\amalg$ on $E.Q$. When there are two transformations f and g transforming the Equivalence query $E.Q$ in two Equivalence queries $E.Q_1$ and $E.Q_2$, it exists one

and only one morphism $I : E.Q \rightarrow E.Q_1 \times E.Q_2$.



2) Diagram explanation: : This diagram is a triangle composed by two triangles. We suppose that there are transformations f and g transforming the Equivalence query $E.Q$ "Who?" in two Equivalence queries $E.Q_1$ "profession?" and $E.Q_2$ "nationality?". Using the product, the Equivalence query $E.Q$ is associated by a unique way to the Equivalence query $E.Q_1 \times E.Q_2$. This Equivalence query is split into Equivalence sub-queries $E.Q_1$ and $E.Q_2$ by using the transformations π_1 and π_2. Each triangle is a commutative diagram: that is to say the computation is supposed to give the same result when performed by each different paths. When the commutative diagram is violated, an error is located in this triangle. By this way, we give a formal semantic to the notion of formal error.

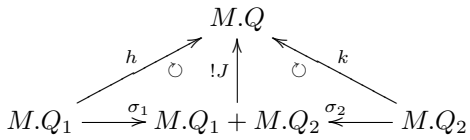
B. Applicative blackbox

The *applicative black box*, like a compiler acts directly on programs in order to produce a program. We define formally an operating system as being a specific category of endofunctors. We call *Membership queries* $M.Q$ the objects of these category. Each Membership query denotes a control on an application component. The Membership sub-queries are combined to produce a Membership query.

The idea is that the applicative blackbox halts or sends a message to the user when the composition of partial Membership queries produces a non-commutative diagram.

This should be given by a prototype: as prototypes are combined by disjunction and composition, Equivalence queries are also combined by sum and composition.

1) Definition: By *applicative blackbox*, we intend a monoidal category for the composition whose objects, called *Membership queries* are endofunctors of $COMP$. On this category, we suppose a sum Σ on $M.Q$. When two transformations h and k transforming two Membership queries $M.Q_1$ and $M.Q_2$ in Membership query $M.Q$ then there is one and only one morphism $J : \Sigma R_i \rightarrow R$.



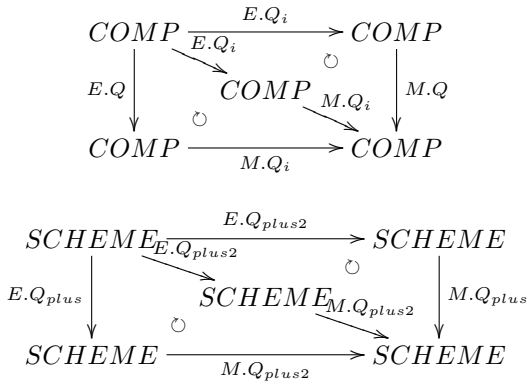
2) Diagram explanation: : This diagram is a triangle made with two triangles. Each node is a Membership query $M.Q$. We suppose two transformations h and k transforming two Membership queries $M.Q_1$ "English" and $M.Q_2$ "Sailor" into the Membership query $M.Q$ "somebody". The Membership query $M.Q$ associated to $M.Q_1 + M.Q_2$ that is composed by the Membership querys $M.Q_1$ and $M.Q_2$ using σ_1 et σ_2. Each triangle is commutative. When the hypothesis of commutation is violated, the error may be located.

C. Junctor

The junctor detects contradictions when the composition of a sub-equivalent query and a Membership sub-query creates a contradiction. In this case, the junctor "disconnects".

When Equivalence queries are canonical forms and Membership queries are prototypes, the junctor disconnects when a prototype satisfies a canonical form and does not satisfy a more general canonical form.

1) Definition: A *Junctor* is represented by the following diagram linking the $E.Q$ of an operating system and the $M.Q$ of an applicative black box.



2) Diagram explanation: :

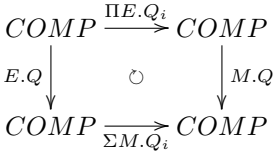
The operating system controls the application-box by composing Membership and Equivalence queries. If $E.Q_i$ is a sub-query "Profession" of $E.Q$ "Who?", and if $M.Q_i$ "sailor" is an answer for $E.Q_i$ "Profession", then $M.Q$ "Somebody" must be an answer to $E.Q_i$ "Profession" and $M.Q_i$ "sailor" must be an answer $E.Q$ "Who?". The adequacy's control imposes that "Sailor" is a partial Membership query of the Equivalence query "Who" and "somebody" is a Membership query of a specific Equivalence query "profession". This Membership query is admitted as correct if the system receives a more specific answer later [14].

This diagram is important because it gives the basis of the formal description of the multi-agent language *Integre*, as discussed in section VII.

D. Cartesian Agent

A Cartesian Agent is an agent able to solve new problems using its operating system and its applicative black box. The new problem is discomposed in Equivalence queries and Membership queries and the new problem's solution is given by a combination of Membership queries. The junctor property warrants local adequacy's control and the "cartesian diagram" warrants that the sum of the partial solutions gives a coherent solution.

1) Definition: By *Cartesian agent*, we intend an operating system and an applicative blackbox such that their Equivalence and Membership queries verify both the jonctor's properties and the following ¿ diagram.



2) Diagram explanation: : A "Cartesian agent" is an "operating system" that "controls" a "blackbox of applications", because combining partial Membership querys gives the same Membership query as the Membership query answering the partial Equivalence queries' product. The cartesian diagram shows a problem's resolution by the cartesian method. To solve a problem, the method is to divide it into solvable subproblems. The solution is then obtained by combining partial solutions.

E. Rational Agent

A rational agent is defined by combining two cartesian agents: a teacher and a learner. Ontology and theory building result from the interaction cycle between agents, which is activated by the learner's dialectical contradiction's management. Now, all the notions required for interactive ontology building are defined. By respecting the interactive learning architecture, a rational agent guaranties a robust learning.

1) Definition: By *Rational Agent*, we intend the composition of two agent's diagrams that respects the following diagram:

$$\begin{array}{ccc} COMP & \xrightarrow{\Pi E.Q_i} & COMP \\ {\scriptstyle E.Q^{learner}}\downarrow & \circlearrowleft & \downarrow{\scriptstyle M.Q^{learner}} \\ COMP & \xrightarrow[\Sigma M.Q_i]{} Theory & \xrightarrow{\Pi E.Q_i} Theory \\ & {\scriptstyle Q^{teacher}}\downarrow \quad \circlearrowleft & \downarrow{\scriptstyle M.Q^{teacher}} \\ & Theory & \xrightarrow[\Sigma M.Q_i]{} Ontology \end{array}$$

2) Diagram explanation: This diagram defines how a rational agent combines two agents: a teacher and a learner. The first one produces an ontology to describe examples from which the second one learns a theory that can be be in contradiction with the teacher's ontology. This process consists in co-building an ontology and a theory.

Let us illustrate it with the previous section's examples where theory and ontology are written in a paraconsistant logic. Given a theory written in paraconsistent logic, the dialectical control activates the ontology revision. This ontology revision implies to modify in the Teacher agent the applications control. This implies to modify the operating system's control. When as in Angluin's paradigm, all these manipulations are only organizing computation, the remaining point is to verify that these manipulations are correct when defined in the category $COMP$.

F. Cognitive relevance of the Rational Agent's formal design

In this section, we formally design a rational agent that can be controled by rules in a paraconsistent logic. Our objective is to maintain the property of robust learning for this agent. This objective is today only a conjecture justified by the fact that our formalism overloads Angluin's one when it defines computerized rational agents that really exists.

Presently, the cognitive performances of our rational agent are the following:

- A rational agent uses a theory and an ontology to control its computation as a "little theorist".
- Up to now, a rational agent is not "a little scientist". Because it is not able to describe the external world by doing experimentations and communicating what it "knows" and what it is "guessing" to others rational agents.

VI. Rational Agent's Supervision By Scientists

In this section, we introduce the human supervision of a rational agent. Here, scientists are coaching the rational agent. Together, they play a scientific game which goal is to build theories and ontologies that enable to predict and explain empirical properties of experiments.

For philosophers [5], two principles act to transform a formal theory into a scientific one:

- *The Reducibility principle* is implicitly related to the ability to reduce a formal proof to a empirical evidence. Let us call EXP an Experimental platform. This principle links a $Theory$ to EXP.
- *The Nominalization principle* is associated to the ability to isolate and name a computation that produces "an empirical visual evidence" in the experimental platform. This principle links $Ontology$ to $Theory$.

These principles give two readings the following diagram that explains the scientist's activity.

$$\begin{array}{ccc} EXP & \xrightarrow{\Pi E.Q_i} & EXP \\ {\scriptstyle E.Q^{scientist}}\downarrow & \circlearrowleft & \downarrow{\scriptstyle M.Q^{scientist}} \\ EXP & \xrightarrow[\Sigma M.Q_i]{} Theory & \xrightarrow{\Pi E.Q_i} Theory \\ & {\scriptstyle Q^{scientist}}\downarrow \quad \circlearrowleft & \downarrow{\scriptstyle M.Q^{scientist}} \\ & Theory & \xrightarrow[\Sigma M.Q_i]{} Ontology \end{array}$$

- The scientists who coach the rational agent are not omniscient.
  - To answer membership queries $M.Q.$, they must do experimentations.
  - To answer equivalence queries $E.Q.$, they must ask for a refutation by the scientific community.

When teachers are scientists, they describe a world; they experiment and publish theories that might be revealed false. The dialog between scientists is required to find and show contradictions, and to progress by resolving them. During this dialog some theoretical errors are detected and the problem's formulation may be revised. Sometimes however, paradoxes occur and activate a major conceptual revision.

Many historical studies show that scientific discovery requires serendipity. Philosophers as Kuhn insist on the fact that errors always have theoretical origins and they emphasize the fact that paradoxes are the source of scientific revolutions. As Plato, they propose dialectic as a philosophical method which uses contradictions to activate a Human revision of a theory. Quine says 'More than once in history, the discovery of a paradox has been the occasion for a major reconstruction at the foundation of thought' [15].

The scientific theory's formation paradox

Even if our current theories in physics, chemistry, biology, or social sciences are sufficient to predict and explain the brain's behavior, a remaining problem is to understand the emergence of the mind's abilities required by scientific activities such as symbolic reasonning, or perpetually revising scientific theories by improvement and refutation.

To design the process of a scientific theory building, philosophers have identified three worlds having their own autonomous behavior. Philosophers are separated in three groups depending on their point of view:

- Conceptualists suppose a primacy of the world of Cognition, which is the brain's intellectual activity, and social activity.
- Nominalists suppose the primacy of the world of mathematical forms, which activity is shown by the development of mathematics.
- Realists suppose the primacy of the Real world, which activity is mainly described by physics, chemistry and biology.

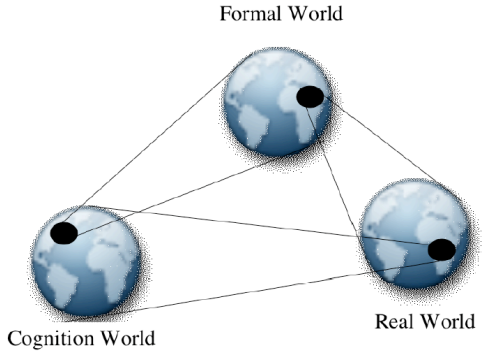


Fig. 6. Penrose's Three worlds'paradox

If we combine, as Penrose [16], these worlds in a circular way, we obtain the following paradox: if a part of human cognition produces formal reasoning as mathematics, if a part of mathematics allows to predict and to explain the real world, and if a part of physics explains human cognition, then "how is it possible that the subjective human's cognitive activity produces formalisms explaining its own mechanisms?". How can a "mundane" scientific activity be able to produce a transcendental knowledge?".

Here, the paradox is the result of chaining three non paradoxical positions together - nominalism, conceptualism and realism.

In our "cognitive informatics" approach, we are nominalists:

- The *nominalization principle* links the world of cognition to the formal world in order to revise the formal model when observing its action in a real world.
- The *reducibility principle* links the formal world to the physical world in order to allow a human visual reasoning.

Then paradoxes are active inside the model and they force adaptation and evolution in a scientific community supervising rational agents.

VII. Concrete framework's implementation

In this section, we present the implementation of E+N in a multi-agent system. Our goal is to show the genericity of the approach from the scientific discovery point of view, as well as for the framework's architecture.

The "Intègre" software platform (http://www.normind.com/integre), developed by the french startup Normind, associates distribution, semantics and coherence to assist the users in the construction of their reference frame for a domain. "Intègre" exploits the projections in various technological fields (dynamic distributed systems, knowledge representation, constraints) and composes them to build, interactively with the user, an adequacy between the observation of the activity in an environment and its definition, in term of semantics and norms.

First, we define the main concepts of this multi-agent system, then we show the relations to our formalism. Finally, we show how we used it to implement E+N.

A. Multi-agent language Integre

This language allows to design a multi-agent system in terms of actions made by agents in environments supervised by institutions.

1) *Environment*: An environment is the problem's resolution space. It's defined by an objective, compound with objects, and populated by agents. It's ruled by at least one Institution and is used by agents to perceive, act, and interact.
2) *Institution*: An Institution is defined by an objective, and it influences at least one environment. It has a normative system to allow it to constraint actions occuring in an environment.
3) *Norms*: Norms are logical rules which constraint an agent's behaviours in an environment. These norms are formulated in a paraconsistent logic.
4) *Action*: An action is attempted by an agent in an environment, and must be validated by an institution.
5) *Agent*: An agent is an entity created to perform an action.

Formal correspondance Let us present the formal correspondance between our formalism and Integre.

1) *Environment*: An environment implements a specific computational category denoted ENV.
2) *Institution*: An institution implements a $CartesianAgent$.
3) *Norms*: a Norm implements the *membership queries* and the *equivalent queries* formulated in a paraconsistent logic.

$$\begin{array}{ccc} ENV & \xrightarrow{\Pi E.Q_i} & ENV \\ {\scriptstyle E.Q}\downarrow & \circlearrowleft & \downarrow{\scriptstyle M.Q} \\ ENV & \xrightarrow[\Sigma M.Q_i]{} & Ontology \end{array}$$

B. E+N's implementation using Intègre

Let us present E+N's implementation in this multi-agent language.

1) *Player*: Let us call "Player" a human playing E+N.
2) *Hidden rule*: A hidden rule is a norm of an institution in charge of an environment.
3) *Private experimentation environment*: A player can create a private experimentation environment supervized by the hidden rule's institution called Nature, to formulate queries.
4) *Nature*: Nature verifies each new sequence created by addition or substitution of a card.
5) *Layout*: The layout displays a view of the user's private experimentation environment.
6) *Working Group*: A working group is a specific environment which institution manages a collaborative activity between players. It allows its members to exchange data, receive pre-publications and share their experiments.
7) *Learner*: A learner is an institution of a working group able to learn from a teacher.
8) *Teacher*: A teacher is an institution of a working group able to teach a learner.
9) *Rational Agent*: A rational agent is the machine formed by the couple learner-teacher that interacts with the player. Each player of a working group is able to play the role of the teacher, that is to say to produce new examples or to modify the ontology.
10) *Oracle*: An oracle is a rational agent of a working group that has published a theory. The oracle is able to predict and explain any experimental result by applying its theory.
11) *Scientific community*: a scientific community is a working group regrouping working groups. It is supervized by an institution to validate identifications, credits, publications, and communication protocols.
12) *Player's actions*: Player's actions are implemented as follow.
    - *Ontology building*: The player can formulate an ontology expressed in conceptual graphs in order to be used by the learning institution to describe the examples.
    - *Experimentation*: An experimentation gives the nature's answer to a rational agent's membership query Me.Q..
    - *Publication/Refutation*: Every member of the community receives a notification when a publication or a refutation occur (an e-mail, for instance). Each player can then verify the coherence of a publication with his own experimentation results, and eventualy produce a counter-example.
13) *Validation of player's actions by the rational agent*: When the player selects a position to play a card, the rational agent predict from its theory four sets of answers: unpredictible cards, cards predicted as correct, cards predicted as not correct, cards predicted as creating contradictions. This informartion allows to estimate the theory in order to know how to modify the ontology, or what kind of example is best suited to improve the theory.
14) This platform allows to easily implement new features which will be usefull to enrich the social game, as the ability to join, quit, or form Working Groups with other players in order to share data, credits, and rational agents. Such a working group is able to fix its private institution. All these new features will extend this game and will allow to study and compare different collective scientific strategies.

On the figure 7, we can see how the main concepts are implemented using Integre.

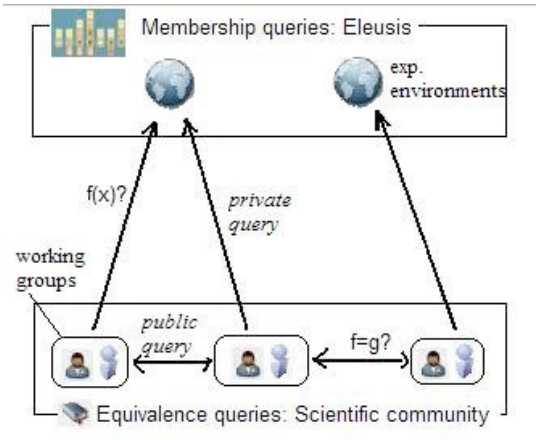


Fig. 7. The queries flow in Eleusis+ Nobel Game

Integre has specific institutions to manage user interaction to web applications and other applications as office.

1) Every player belongs to a scientific community which is also constrained by norms describing a communication protocol which fix how and when a player can publish or refute, and the informations that must be visible on the publication (the ontology used, a valid sequence,...).
2) The scientific community manages storage and access to the published documents, and informations about the game (credits, number of players...).

This implementation emphasizes that an information system can be realized by a rigid architecture that manages permanently evolving processes.

The *Rational Agent machine* is composed by a teacher having the role to know how to realize interesting applications and a learner whose role is to know how to combine its own applications. Combining them creates a machine that is always learning.

In this implementation, the player delegates to the rational agent the theory formation which is a real innovation: more than a computer, it is an *arguer* able to *argue* its theories.

The ontology building method's efficiency comes from the dynamics between an ontology revised by a man and a theory built by a machine that shows to the human what bias comes from incompletness of the examples. This is clearly shown on E+N experimentations in which different working groups having different experimentation stategies produce different incompatible theories.

VIII. Discussion and perspectives

Let us discuss our contribution to cognitive informatics.

Scientists study now a class of complex problems that have no a-priori theory or model. They experiment, publish and progress in the understanding of their problems. Generally, experts are not omniscient and with their competences, they create deep but ”regional” ontologies. Our methodology allows them to share their ontologies with experts from other domains.

Jon Doyle [17] has published one of the first work about rational psychology. As our rational agent has the ability to organize computations in order to stabilize its current state, it represents an attempt to give a foundation to such a type of psychology. It does it by focussing on the communication with users, reasoning on contradictions and participating to the description of a world. All these behaviours don't use a representation of itself or others. They don't give any value to the produced information, they don't interpret errors, they don't acknowledge the existence of each others, they don't act on reality. For all these reasons, a rational agent's psychology is very ingenuous, it doesn't act directly on its world, but via a human agent. It doesn't interpret other's behaviour.

In this context, another interesting research direction would be to study the types of cognitive disorders of a rational agent. Would an ambivalent teacher, giving contradictory informations, provoke an identity disorder? Could a rational agent be stuck by contradictions in the interaction cycle, being unable to build the capacity to distinguish itself from the world?

With this first version, we could already observe how misfunctions occur, such as disorders of identity or of self-consciousness, of self-affirmation via contradiction, of language formation or world description.

Finally, this version can be extended by giving a formal semantic to the relation between rational agents. This formalism would allow a rational agent to determine their social behaviours. In such a context, our objective is to formalize the spiritual automata described by Spinoza in ethics [18].

- Agent build their identity by teaching: each machine stabilizes itself by a *personal* way.
- The learner-teacher interaction gives them a consciousness of their behavior.
- Agents show their internal emotions when they locate the source of contradictions.
- Today they have no consciousness of others' actions: the emergence of an apparently cognitive behavior occur in a social game when they interact.
- They have no ”social emotion”: when they make a mistake, they don't make others responsible for what is happening to them.

IX. Conclusion

In this article, we propose a conceptual framework based on rational agents. These machines, respecting Angluin's ”learning from different agents” paradigm, learn how to manage the applications on behalf of the users.

Since to teach a rational agent is a way to build an ontology free of contradictions, we propose an effictive way to assist scientists in their conception and revision of ontologies.

We present the experimental framework consisting of a scientific game E+N that has been developed in order to embody this new approach in assisting scientific discovery.

Rational agents have important cognitive faculties, as identity, a consciousness of their behavior, a dialectical control of theoretical contradictions in a learned theory respecting a given ontology, and the aptitude to propose ontology revisions.

Satosi Watanabe [19], a pioneer in Artificial Intelligence, inspired himself from a Confucius' aphorism when he affirms: ”an intelligent machine cannot be a slave”. The Rational agents we present have the autonomy to manage their own applications. We can instruct them since they are able to do autoprogramming.

Cavaillès [5] establishes a correspondance between Mind consciousness formation and a lively mathematic having a long history of conceptual transformations in order to overcome paradoxes. We think that Cognitive Informatics participates to this vision. Computers help Humans to produce useful abstractions to predict and explain the complex systems that we are and in which we live. They are *rational mirrors* for human minds.

Acknowledgment

We thank professor Daniel Guin for his support on the formal aspects. This work has been partially supported by the Information Society Technologies (IST) programs: ORIEL (Online Research Information Environment for the Life Sciences, IST-2001-32688). We also thank the Normind Company for lending us Integre.

References

[1] D. Angluin and M. Krikis, “Learning from different teachers,” *Machine Learning*, vol. 51, pp. 137–163, 2003.
[2] N. Littlestone, “Learning quickly with irrelevant attributes abound. a new linear threshold algorithm,” *Machine Learning*, vol. 2, pp. 285–318, 1988.
[3] M. Gardner, “Mathematical games,” *Scientific American*, June 1959.
[4] D. Chavalarias, “La thèse de popper est-elle réfutable ?” CREA - CNRS/Ecole Polytechnique” Memoire de DEA, 1997.
[5] J. Cavaillès, *Sur la logique et la théorie de la science*. Librairie Philosophique J. VRIN, 1997.
[6] D. Angluin, M. Krikis, R. Sloan, and G. Turan, “Malicious omission and errors in answers to membership queries,” *Machine learning*, vol. 28, pp. 211–255, 1997.
[7] D. Angluin, “Queries and concept learning,” *Machine Learning*, vol. 2, no. 4, pp. 319–342, 1988.
[8] N. H. Bshouty, S. A. Goldman, T. R. Hancock, and S. Matar, “Asquing question to minimise errors,” *Journal of computer and system sciences*, vol. 52, pp. 268–286, 1996.
[9] D. Angluin, “Queries revisited,” *Theoretical Computer Science*, vol. 313, pp. 175–194, 2004.
[10] G. M. da Nobrega, S. A. Cerri, and J. Sallantin, “A contradiction-driven approach to theory information: Conceptual issues pragmatics in human learning, potentialities,” *Journal of the Brazilian Computer Society*, vol. 9, no. 2, pp. 37–55, nov 2003.
[11] N. C. A. da Costa, *Logiques classiques et non classiques. Essai sur les fondements de la logique*, ser. Culture scientifique. Masson, 1997.
[12] N. C. A. da Costa and J.-Y. Beziau, “La logique paraconsistante,” in *Le concept de preuve à la lumière de l'intelligence artificielle*, J. Sallantin and J. J. Szczeciniarz, Eds. Presses Universitaires de France, 1999, pp. 107–117.
[13] M. Liquière and J. Sallantin, “Structural machine learning with galois lattice and graph,” in *ICML'98, Madison, Wisconsin, July, 1998*, 1998, pp. 305–317.

[14] E. Castro, J. Sallantin, and S. Cerri, "Misunderstanding detection using a constrainted based mediator," in *ALCAA Agents Logiciels Coopération Apprentissage & Activité Humaine, Biarritz, France, October 2000*, 2000.
[15] M. Clark, *Paradoxes from A to Z*. Routledge and Kegan Paul, 2002.
[16] R. Penrose, A. Shimony, N. Cartwright, and S. Hawking, *The Large, the Small and the Human Mind*, R. Penrose and M. Longair, Eds. Cambridge University Press, 2000.
[17] J. Doyle, "Reasoned assumptions and rational psychology," *Fundamenta Informaticae*, 1994.
[18] Spinoza, *Ethique*. Éditions du Seuil, 1999.
[19] S. Watanabe, *Knowing and Guessing*. John Wiley and Sons, 1969.

Jean Sallantin

Jean Sallantin is Director of Research in CNRS. His early work deals with Machine Learning and their applications in different fields of Science as Geophysics, Molecular Biology and Law. He his currently working on computational philosophy, mainly on the cognitive and computational aspects of scientific discovery.

Christopher Dartnell

Christopher Dartnell received a Master degree in Computer Science in 2003 from Université de Montpellier II, France and is currently working as a Ph.D. student. The topic of his Thesis deals with supervision of complex systems, and assistance to scientific discovery. He develops a software platform for cognitive and computational experimentations of scientific discovery, named E+N for Eleusis+Nobel.

Jacques Divol

Jacques Divol received a engineer degree in Computer Science in 2003 from CNAM, France. His early work deals with Knowledge Acquisition by teaching a learning machine. He has worked as research engineer on the ORIEL (IST-2001-32688) project funded by the European Commission. He is currently working as research developping a software tool for Interactive Ontology Building, named WEBRA for Web Rational Agent.

Patrice Duroux

Patrice Duroux received a Ph.D. degree in Computer Science in 1999 from Université de Montpellier II, France. His early work deals with computational linguistics and computational aspects of the theory of categories. He is currently working as a research engineer on the ORIEL (IST-2001-32688) project funded by the European Commission. He is developing a software tool to integrate resources in bio-informatics, named BiODS.