The G.O.A.L. Approach

A Goal-Oriented Algebraic Language

Dantan Jérôme[1,2], Pollet Yann[1] and Taibi Salima[2]
[1]*CEDRIC, CNAM, My Street, Paris, France*
[2]*LAMSAD-Agri'terr, Esitpa-APCA, Mont-Saint-Aignan, France*
jdantan@esitpa.org, yann.pollet@cnam.fr, staibi@esitpa.org

Keywords: Goal, Algebra, Semantic Web, Ontology, Semantic Web Service.

Abstract: In a global context of sharing information, such as Big Data and cloud computing paradigms, researchers are developing many means to deal with new data models and algorithms. However, the development and reuse of these ones is complex because of the heterogeneity of environments, data formats and contexts of use all around of the world. That's why a way to share and reuse algorithms and treatments through a common formalism is needed, for both machines and computers. The ultimate goal of our work is to provide a collaborative platform for not only experts but also machines automatically develop, reuse and chain treatments, computations and models. For this, we rely on a goal-oriented approach which is associated with the Semantic Web, to establish a common formalism to design models for worldwide researchers. In this article, we propose the formalization of our approach thanks to an algebra which is linked to the Semantic Web standards. Finally, we provide a high-level language dedicated to both computers and experts, illustrated with examples that are linked to the agriculture domain.

1 INTRODUCTION

Assessing the sustainability of human activities is becoming a worldwide major concern. In this paper, we consider examples that belong to the agriculture domain such as agronomy, ecology, economy, sustainability modeling, etc. Researchers in the agriculture domain do not only seek to assess and improve the performances of farms, but also their sustainability, through many indicators. The development and computation of such indicators require being expert in several areas. Indeed, some aspects to be treated may be e.g. soil biology, animal welfare, waste management, economic durability, sales, etc.

For this, researchers from all over the world compute data from many sources, both local and from the Internet, that cannot be used directly, e.g. data from RDBMS (Relational DataBase Management Systems) and complex models. Consequently, many computer programs and database queries have been developed for interperting data. However, they are specific to each data source but experts need to reuse the code already written e.g. to extract data from other sources. That's why they need a common platform not only for experts design support but also for computers to assist and/or automate the development and reuse of treatments, models, etc. Large amounts of data and treatments that are stored on the Internet and from disparate sources must be computed, according to a common formalism that we propose in this paper.

The remainder of this article is structured as follows: in section 2 we provide a state of the art. In section 3, we present the features of our approach. Then, in sections 4 and 5, we respectively provide definitions and algebra we have defined. In section 6, we introduce he high-level language we have developped, illustrated with examples in agriculture. Finally we conclude in section 7.

2 STATE OF THE ART

Our approach involves several areas in software engineering. We need to index treatments and data though a common formalism. This common formalim must be understandable for both humans and machines. We have identified two

complementary approaches that may meet our expectations. First of all, the goal-oriented approaches are usually used to formalize a problem thanks to hierarchies of goals. Then the Semantic Web approaches make models understandable by both machines and humans, with the objective to reuse software components.

2.1 The goal-oriented approaches

The existing goal-oriented approaches are intended to formalize a problem by defining the steps and components of software design in terms of goals. A goal is by definition a target to reach, part of the Requirement Engineering. The way the goal is achieved is represented by a decomposition of the goal into sub-goals, which "consists in identifying goals and refining them into sub-goals until the latter can be assigned as responsibilities of single agents such as humans, devices and software" (Letier, E., 2001). According to (Lapouchnian, A., 2005), "The main measure of the success of a software system is the degree to which it meets its purpose. Therefore, identifying this purpose must be one of the main activities in the development of software systems". In KAOS (Knowledge Acquisition in autOmated Specification or Keep All Objects Satisfied), users are able to structure the goals: each goal, except the roots, is justified by at least another goal that explains why the goal was introduced in the model and each goal, except the leaves, is decomposed in sub-goals, describing how the decomposed goal can be reached.

More recently, (Guzelian G., Cauvet C., Ramadour P., 2004) describe the modelling of information systems added to the methods of object design, with problems "expressed as goals to be reached and the information system is the result of a process of meeting these goals". Moreover, they present their approach by defining meta models including decompositions of goals into refined sub-goals, formalized thanks to UML. A model, allowing to prioritize and organize components with a set of decision trees and a system of components reuse are proposed. However, neither algebra nor high level language for goals assessement is presented.

Finally, the goal-oriented approaches are used in many fields of software engineering: Requirement Engineering, business modeling, specification of reusable components, definition of user models and development of interactive systems as information systems on the Web, agent systems, information retrieval systems, etc (Guzelian G., Cauvet C., Ramadour P., 2004).

2.2 The Semantic Web

The Semantic Web (SW or Web of Data) is an extension of the existing Web, providing access to computers to structured collections of information and sets of inference rules that they can use to achieve automated reasoning (Berners-Lee, T., Hendler, J., and Lassila, O., 2001). For this, the Semantic Web is based among others on representations of human knowledge based on domain ontologies and semantic web services, which are autonomous computing entities that compute semantically indexed data.

According to (Castellani, S. et al, 2011), Ontologies offer significant benefits to collaborative service-oriented systems, such as interoperability and reusability. They are usually used as an index to retrieve specific data (Garcia, R., Celma, O., 2005), to infer new knowledge (W3C, 2012) (Berners-Lee, T., Hendler, J., and Lassila, O., 2001), to semantically annotate multimedia data (Castano, S. et al, 2007), to find out Web Services automatically (Martin, D et al, 2007), or to match knowledge with other knowledge for a more general purpose (Cruz, C. and Nicolle, C., 2011). Furthermore, the Web Ontology Language (OWL) provides model flexibility, explicit representation of semantics, out-of-the-box reasoning and freely available background knowledge (Castellani, S et al, 2011).

To manage knowledge, the Semantic Web Services (SWS) are based on semantic description frameworks for Web Services. The composition of SWS is the automated processing of Web services autonomously simple to complex automated processes. Then, a "generic inference mechanism shall be developed for handling SWS" (Charif, Y., Sabouret, N., 2006). Various technologies exist, such as OWL-S (W3C, 2004), WSMO (W3C, 2005), METEOR-S (Sheth, A. P., Gomadam, K., Ranabahu, A., 2008), IRS-III (Domingue, J. et al, 2008), etc. The main drawbacks of these approaches are that the user must be a computer specialist, whereas the services composition solutions are intended to help ordinary users in the web, and some manual steps must be performed by the user (Charif, Y., Sabouret, N., 2006).

2.3 Combining the approaches

Our goal is not only to analyze the requirements of a system. Our platform has also to find the ad hoc compositions of treatments solving both users' and computers' requirements, to formalize models,

algorithms and data to assist or automate their reuse, to assess them, and consequently to be both worldwide researchers and machine understandable. In (Dantan, J., Pollet, Y., Taibi, S., 2012), two kinds of virtual properties have been defined, as an extension of OWL (W3C, 2012): these virtual properties, attached to core OWL ontology classes, may be hierarchically organized goals that have to be evaluated. We provide a goal-oriented model that is linked to the Semantic Web properties. We combine a core OWL ontology extended by virtual properties with a goal ontology that contains hierarchically organised goals and sub-goals. Indeed, we consider that the assessment of a higher-level goal is achieved when the assessement of its sub-goals, i.e. the goals at a more specific level, are achieved. At the below, the goals are raw data extraction. For this, we first formalize a rigorous algebra based on well defined operators and then we provide a common high-level language for both users and computers to automatically develop, reuse and chain the treatments, computations and models.

3 OVERVIEW

We propose a goal-oriented model where each goal is attached to both the domain ontology, as a "goal property" and to a semantic goal-oriented service, according to the following procedures:

(1) A core OWL ontology which is part of the Semantic Web is used to model the considered domain. We attach goal assessments as classes' properties of the selected core ontology, which consists of an extension of core ontology.

(2) Formalization of data models, algorithms and non-functional properties in a common formalism for expressing goals and their assessment, by defining an algebra and a high level language that is both machine and human understandable.

(3) Linking with the Semantic Web. Every data models and algorithms are semantically indexed and each goal property is defined thanks to an a URI (Uniform Resource Identifier). Thus, each abstract service is considered as both a URI and a property of classes from the core ontology. Their concrete services are considered as their instances (objects).

We therefore structure models, algorithms and data by using the ontology of goals that are formulated either by the machine or the user who is not a computer scientist. The various solutions result from the various feasible compositions of goals into sub-goals.

4 DEFINITIONS

In this section, we define the notions of goal, goal property and goal-oriented service and finally we illustrate our meta-model.

4.1 Goal

We define a goal as a semantically indexed quantity to evaluate, that is attached to one or more elements of real life., e.g. the soil quality of a parcel, the sustainability of a farm, the sales of a farm, etc. There are goals defined by composition, i.e. goals that can be composed into sub-goals, which must be reached for the higher level goal to be satisfied. In other words, a process defined by composition of sub-goals ensures a higher level goal.

4.2 Goal property

Each goal is linked to a goal property that belongs to an ontology. A goal property is the assessable property that is attached to an OWL ontology class. In other words, we define a goal property as the projection of an ontology class to an assessable goal property, such as a datatype property, or a virtual property (Dantan, J., Pollet, Y., Taibi, S., 2012) that may be either quantitative or qualitative data.

For example, in the agriculture domain, the evaluation of the economic durability of a farm is a goal. The "economic durability" goal property may be assessable via the projection of the "farm" class to the "economic durability" goal. The "economic durability" goal property may be assessed thanks to (1) simple datatype properties such as the sales or the surface of a farm, and (2) other goal properties that are more complex, such as the soil quality which may be assessed by several algorithms. We therefore define two main types of goal properties:

(1) Complex: complex processes using quantitative or qualitative data (e.g. sustainability computations, complex statistical treatments, etc).

(2) Data: processing of raw data (e.g. subscription to events such as live results of soil analyses, data extraction, and aggregate operators).

Finally, a composition of goal properties is a set of goal properties that are linked thanks to operators such as "and", "or", and aggregate operators, which leads to a higher level goal.

4.3 Goal-oriented service

A goal-oriented service is a computer entity which is the mean both to reach a goal and to assess a goal

property, with inputs and outputs. The function associated to a goal-oriented service is the model, treatment, algorithm or simply a datatype property that evaluates the associated goal property.

The inputs of a goal-oriented service exactly match the outputs of the sub-goal services that compose them. Indeed, the principle of approached matching is not able to provide a service which actually meets to the need formulated (Pollet, Y., 2010) (Dantan, J., Pollet, Y., Taibi, S., 2012). We therefore define compositions of goal properties by enabling either non computer scientist users or computers to manage compositions of goal properties in a declarative way. To do this, the experts of the considered domain first work in an generic or abstract way, and the technical details and data restrictions of concrete services are not known. The discovery and composition of services are automatically performed. Each generic goal-oriented service is identified thanks to a URI. The classes of the ontology are URIs and their instances are the concrete implementations for a particular set of data.

4.4 The meta-model

The meta-model, which illustrated in figure 1, contains three layers: (1) the ontology layer that is the core ontology, extended with the goal properties; (2) the goal layer that is the hierarchy of goals; their associated goal properties extend the core domain ontology; (3) the goal-oriented services layer that contains the goal-oriented services.

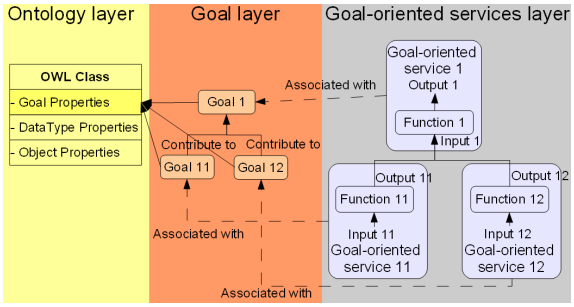

Figure 1: The meta-model.

5 ALGEBRA

We define an algebra that includes operators, which operate on both ontology classes and properties. We have built algebraic expressions to express a goal in a formal way.

In the definitions that will follow; let C, C_1, C_2, …, C_n be ontology classes and let G, G_1, G_2, …, G_n be one of their respectively attached goal properties.

In the examples that will follow:

- Let "farm", "parcel" be ontology classes.
- Let "sustainability", "sales" "productivity" ,"economicDurability", "area", "culture", "ISO14000-sales", "ISO14000-productivity" be goal properties attached to both the "farm" class and classes that are contained in the farm class.
- "sales" and "productivity" goal properties contribute to the "economicDurability" higher level goal property.
- Let "contains" be the contains object property attached to the farm class.

5.1 Projection of a class to a property

We define a projection as a restriction of an ontology class to one of its properties. The inputs of the projection operator are either an ontology class or a property of this class, which can be: (1) either a goal property. In this case, the result of the projection is an assessable goal property or (2) an object property. In this case, the result of the projection is a set of object properties.

5.1.1 Projection to a goal property

The projection of a class to a goal property is denoted: $\pi_G C$, where C is an ontology class and G is the goal property to which the class is restricted e.g. the projection of the farm class to its sustainability goal property. It is denoted: $\pi_{sustainability}\ farm$. The result is the assessable "sustainability" goal property, attached to the farm class.

5.1.2 Projection to an object property

The projection of an ontology class to an object property is denoted: $\pi'_O C$, where C is an ontology class and O is the object property to which the class is restricted. The projection of the farm class to its "contains" object property is denoted: $\pi'_{contains}\ farm$. The result is the set of classes that the farm class contains, e.g. parcels, barns, etc.

5.2 Conjunctions and disjunctions

The goal-oriented model we propose may express various solutions to assess a given high-level goal.

These goal properties are basically described thanks to "and" and "or" logic operators, by building decisions trees, thus enabling the assessment of a high-level goal property and defining a complete process of goals/sub-goals expression. By extension, in our model, each conjunction/disjunction of goal property meets one-to-one to their conjunction/disjunction of goals.

(1) The "AND" conjunction. A goal property is achieved when its preconditions are achieved. In fact we define the achievement of a high level goal property as a "and" conjunction of lower level goal properties, denoted: $\pi_{(G_1)}C_1 \wedge \pi_{(G_2)}C_2$. For instance, the conjunction of the "sales" and "productivity" farm goal properties is denoted: $\pi_{sales}\, farm \wedge \pi_{productivity}\, farm$. We will see that such conjunction of goals may reach a higher level goal.

(2) The "or" disjunction: when several scenarios of conjunctions or compositions exist, we express the alternatives through the "or" conjunction. It means that each alternative conjunction or composition below is interchangeable to reach the same goal. Two alternative compositions are denoted: $\pi_{(G_1)}C_1 \vee \pi_{(G_2)}C_2$. So $\pi_{(G_1)}C_1$ and $\pi_{(G_2)}C_2$ are alternative compositions.

5.3 Chronology

It is possible to define the chronology of sub-goals and goals, to link goals to sub-goals explicitly thanks to a temporal conjunction. This may enable the user, while expressing a request, to suggest its preferences to the composition engine and to other users. The composition engine of services can achieve the higher level goal property only if its precondition goal-properties (sub-goals), have been achieved. So, a composition of sub-goals $(\pi_{(G_1)}C_1 \ldots \pi_{(G_n)}C_n)$ leading to a higher level goal $\pi_G C$ is denoted as follows:

$$(\pi_{(G_1)}C_1 \ldots \pi_{(G_n)}C_n) \rightarrow \pi_G C \qquad (1)$$

For example, the conjunction of the farm goal properties mentioned just before, leading to the farm economic durability are denoted as follows:

$$\pi_{sales}\, farm \wedge \pi_{productivity}\, farm \rightarrow \pi_{economicDurability}\, farm \qquad (2)$$

5.4 Aggregates

The standard relational algebra aggregates are: sum, average, minimum, maximum and count. Each aggregate has a list of assessable goals for input and has an assessed goal for output. An aggregate function is denoted: $AGG(\pi_G C)$, where AGG is one of the following aggregate operators: SUM, AVG, MIN, MAX and COUNT.

5.5 General example

The following example is a combination of the cases already mentioned. Suppose that the sum of the productivities of the classes that compose a farm is "equivalent" to the whole productivity of a farm.

The conjunction of the farm goal properties mentioned just before, with the sum of productivities alternative conjunction about the classes that are contained into the farm, leading to the farm economic durability are denoted as follows:

$$\pi_{sales}\, farm \wedge (\pi_{productivity}\, farm \vee \mathrm{SUM}(\pi_{productivity}(\pi'_{contains}\, farm))) \rightarrow \pi_{economicDurability}\, farm \qquad (3)$$

Note that the use of the SUM operator involves that we take into consideration the whole classes that may be included in the farm.

5.6 Restriction

The restriction operator selects a composition of goal properties for given sets of goal properties. So, the restriction is performed through a restriction expression expressed thanks to a particular conjunction of assessed goal properties. The restriction of a composition of goal properties is denoted: $\sigma_R(\wedge C_i)$, where $(\wedge C_i)$ is a composition of goals and R is the restriction expressed thanks to a conjunction of goals, as defined above. The restriction operator requires the evaluation of the composition of goal properties, e.g. the restriction of the economic durability composition, restricted to the farms that belong to the Normandy area and that cultivate potatoes is denoted as follows:

$$\sigma_{(\pi_{(area=\,normandy)}.farm \;\wedge\; \pi_{(culture=\,potatoes)}.farm)} (\pi_{sales}\, farm \wedge \pi_{productivity}\, farm \rightarrow \pi_{economicDurability}\, farm) \qquad (4)$$

5.7 Operator and property overload

Goal properties and aggregate operators can be overloaded either by taking into account either theoretical implementations or new input/output data. They are not theoretically implemented yet but may overload the existing goal properties, e.g. by inheritance, or by aggregate operators overloading, taking account of specific computing methods or new data types.

(1) The overload of a goal property is specified by adding a "+" as exponent: $\pi_{(G^+)}C$.

(2) The overload of an aggregate operator is specified by adding a "+" as exponent: $AGG^+(\pi_G C)$.

Examples: (1) the "agronomicSustainability" parcel goal property may be overloaded by the "agronomicSustainability" goal property of a farm. (2) The "MAX" aggregate operator that extracts the maximum from a set of quantitative values may be overloaded to extract the maximum from a set of qualitative values, by implementing a new way to compute them.

5.8 Inheritance

Inheritance is a specialization of a goal that is different from a "restriction" which restricts a goal property to a particular data set, as a geographical area e.g.. inheritance rather enables to declare a new goal from another specializing it "functionally", e.g. based on a particular standard. Sub-goals that compose a higher level goal may correspond to several computations in addition to potential restrictions on a data set. The inheritance of a goal property from another goal property can be explicitly formulated by new operations on goal properties. It is denoted: $\pi_G C \leftarrow \pi_{(G')} C$, where G' is the goal property inherits from G.

For example, the "ISO14000-sustainability" farm goal property inherits from the "sustainability" farm goal property because it specializes it by achieving their computations according to ISO14000 standards which defines standards for environmental management. The inheritance of farm ISO14000-sustainability from farm sustainability is denoted as follows:

$$\pi_{sustainability}\, farm \leftarrow \pi_{\text{ISO14000-sustainability}}\, farm \qquad (5)$$

5.9 Goal-equivalence

Two conjunctions of goal properties are called equivalent if they both contribute to the same goal property whatever restrictions that can be applied to them. The goal-equivalence of two goal properties is denoted: $\pi_{(G_X)}C = \pi_{(G_Y)}C$, where G_X and G_Y are two goal properties from two goal-equivalent compositions. The goal-equivalence relation is an equivalence relation (reflexive, symmetric , transitive). The following conjunctions are goal-equivalent:

$$\begin{gathered}((\pi_{(ISO14000-sales)}\, farm) \wedge (\pi_{(ISO14000-productivity)}\, farm)) \\ = \\ (\pi_{(ISO14000-sales)}\, farm \wedge \text{SUM}(\pi_{(ISO14000-productivity)}(\pi'_{contains}\, farm)))\end{gathered} \qquad (6)$$

Explanation: one of the criteria of ISO 14000 is just to retrieve all the data related to their goals that are available and consequently the sub-goals are theoretically always satisfied. So, they both satisfy the higher-level goal property: $\pi_{(ISO14000-economicDurability)}\, farm$, regardless of the restrictions that can be applied to them.

5.10 Goal-sub-equivalence

Two conjunctions of goal properties are called goal-sub-equivalent if they both contribute to the same goal property, restricted to a particular conjunction of assessed goal properties. The goal-sub-equivalence of two goal properties is denoted: $\pi_{(G_X)}C \simeq \pi_{(G_Y)}C$, where C is an ontology class, G_X and G_Y are two goal properties from two equivalent compositions. The goal-sub-equivalence relation is an equivalence relation (reflexive, symmetric, transitive). The following conjunctions are goal-sub-equivalent:

$$\begin{gathered}\sigma_{(\pi_{(area=normandy)}farm \wedge \pi_{(culture=potatoes)}farm)}(\pi_{(sales\ farm)} \wedge (\pi_{(productivity)}\, farm)) \\ \simeq \\ \sigma_{(\pi_{(country=normandy)}farm \wedge \pi_{(culture=potatoes)}farm)} \\ (\pi_{(sales)}\, farm \wedge \text{SUM}(\pi_{(productivity)}(\pi'_{contains}\, farm)))\end{gathered} \qquad (7)$$

Explanation: in the area of Normandy all potatoes fields have all the data related to their goals that is available and consequently the sub-goals are theoretically always satisfied. So, they both satisfy the higher-level goal property, for farms that cultivate potatoes in Normandy:

$$\sigma_{(\pi_{(area=normandy)}farm \wedge \pi_{(culture=potatoes)}farm)}(\pi_{(ISO14000-economicDurability)}\, farm) \qquad (8)$$

5.11 Priorities of operators

Parentheses may prioritize the operators. However, this is the list of operators in decreasing order of priority: Projection, And, Or, Then, Restriction. The priorities for aggregate, overload, inheritance, goal-equivalence and goal-sub-equivalence are not applicable because they must be explicitly defined with parenthesis.

6 THE G.O.A.L

To compose the goals, their attached goal properties and goal-oriented services, we have specified a both

user and machine understandable high-level language. We have called this language Goal Oriented Algebraic Language (G.O.A.L); it relies on the algebraic operators we have just defined.

6.1 Syntax

Table 1 summarizes the syntax of both our algebra and our high-level language operators.

Table 1: algebraic and GOAL notation .

Operator name	Algebraic operator	GOAL notation
Projection	π	. (dot)
Aggregates	SUM AVG MIN MAX COUNT	SUM AVG MIN, MAX COUNT
And	$\wedge$	AND
Or	$\vee$	OR
Then	$\rightarrow$	THEN
Restriction	σ	WHERE
Overload	$^{+}$ exponent	SUPER
Inheritance	$\leftarrow$	EXTENDS
Goal-equivalence	$=$	EQUIVALENT_TO
Goal-sub-equivalence	$\simeq$	SUBEQUIVALENT_TO

6.2 Examples

Here are some examples linked to agriculture.In the examples that will follow:

- Let “farm”, “territory” be ontology classes.
- Let “sustainability”, “sales”, “productivity”,“economicDurability”, “metalRate” “pesticideRate”, “country”, “culture”, “ISO14000-sales”, “ISO14000-productivity”, “ISO14000-sustainability” be goal properties attached to the “farm” class.
- Let “confidentiality” and “reliability” be non-functionnal properties.
- “sales” and “productivity” goal properties contribute to the “economicDurability” higher level goal property.
- Let “contains” be the contains object property attached to the farm class.

6.2.1 Projection, aggregate, conjunction and disjunction operators

The generic sustainability property of the farm class can be expressed as follows:

```
(farm.sales AND (farm.productivity OR
farm.contains.?).productivity)
THEN farm.economicDurability
```

6.2.2 Complex goal-oriented services

Some goal properties are expressible through more complex statistical operators, e.g. the landscape of an area may be inferred from a decision tree whose leaves are sub-goals, such as metal rates and pesticide rates:

```
DECISION_TREE(AVG(territory.metalRate) AND
AVG(territory.pesticideRate))
THEN territory.landscape
```

6.2.3 Restriction

The concrete services are expressed by adding a “WHERE” restriction clause with functionnal properties, following the generic term of the goal property. Next, a non-functionnal clause such as QoS can be specified by adding a “WITH” restriction clause, e.g.:

```
((farm.sales AND (farm.productivity)
THEN farm.economicDurability)
WHERE (farm.country=France AND
farm.culture=potatoes)
WITH (farm.sales.confidentiality=high AND
farm.productivity.reliability=medium)
```

6.2.4 Goal-equivalence

Two conjunctions of goal properties are called equivalent if they both contribute to the same goal property whatever restrictions that can be applied to them, e.g.:

```
(farm.ISO14000-sales AND farm.ISO14000-
productivity)
EQUIVALENT_TO
(farm.ISO14000-sales AND SUM
(farm.contains.?)).ISO14000-productivity
```

The compositions below are equivalent because one of the criteria of ISO 14000 is just to own all the data related to their sub-goals, that are theoretically always satisfied.

6.2.5 Inheritance

The "ISO14000-sustainability" farm goal property inherits from the "sustainability" farm goal property:

```
farm.ISO14000-sustainability
EXTENDS farm.sustainability
```

6.2.6 Goal-sub-equivalence

Two conjunctions of goal properties that are goal-sub-equivalent are denoted as follows:

```
((farm.sales AND farm.productivity)
THEN farm.durability)
SUBEQUIVALENT_TO
((farm.sales AND SUM
(farm.contains.?).productivity) THEN
farm.durability)
WHERE (farm.area=normandy AND
farm.culture=potatoes)
```

The compositions below are equivalent because, in the area of Normandy, all data fields of potatoes required for the satisfaction of all the sub-goals of these compositions are available.

7 CONCLUSION

In a nutshell, we have presented the association between a goal-oriented approach and the Semantic Web. To formalize our approach, we have proposed an algebra which is linked to OWL ontologies classes and properties. Then, we have provided a high level language called Goal-Oriented Algebraic Language (GOAL), dedicated to both computers and experts and based on our algebra. Finally, we have illustrated our approach thanks to examples, linked to agriculture. Future challenges are to provide an ontology and actual use cases about the goal-oriented Web services.

REFERENCES

Letier, E., 2001. *Reasoning about Agents in Goal-Oriented Requirements Engineering*, Phd Thesis. Université Catholique de Louvain, Dépt. Ingénierie Informatique, Louvain-la-Neuve, Belgium.

Lapouchnian, A., 2005. *Goal-Oriented Requirements Engineering: An Overview of the Current Research.* Depth Report, University of Toronto.

Guzelian G., Cauvet C., Ramadour P., 2004. Conception et réutilisation de composants : une approche par les buts. *In INFORSID 2004*, 179-174.

Berners-Lee, T., Hendler, J., and Lassila, O., 2001. The Semantic Web. Scientic American, 284(5):35-43.

Castellani, S. et al, 2011. A knownledge-based system to support legal case construction. *In Knowledge Engineering and Ontology Development conference.* Paris, France: 15-27, may 2001.

Garcia, R., Celma, O., 2005. Semantic Integration and Retrieval of Multimedia Metadata. *In 4rd International Semantic Web Conference*, Galway, Ireland.

World Wide Web Consortium (W3C), 2012. *OWL 2 Web Ontology Language Document Overview (Second Edition) http://www.w3.org/TR/owl2-overview/.*

Castano, S. et al, 2007. Ontology Dynamics with Multimedia Information: The BOEMIE Evolution Methodology. *In International Workshop on Ontology Dynamics (IWOD) ESWC 2007 Workshop.* Innsbruck, Austria.

Martin, D et al, 2007. Towards Semantic Annotations of Web Services: OWL-S from the SAWSDL Perspective. *Proceedings of OWL-S Experiences and Future Developments Workshop at ESWC 2007.* Innsbruck, Austria.

Cruz, C. and Nicolle, C., 2011. A graph-based tool for the translation of xml data to OWL-DL ontologies. *In Knowledge Engineering and Ontology Development (KEOD) IC3K 2011 conference.* Paris, France, pp. 361-364.

Charif, Y., Sabouret, N., 2006. An Overview of Semantic Web Services Composition Approaches. *Electronic Notes in Theoretical Computer Science.* Volume 146, Issue 1, 24 January 2006, 33-41.

World Wide Web Consortium (W3C), 2004. *OWL-S: Semantic Markup for Web Services* http://www.w3.org/Submission/OWL-S/.

World Wide Web Consortium (W3C), 2005. *Web Service Modeling Ontology (WSMO)* http://www.w3.org/Submission/WSMO/.

Sheth, A. P., Gomadam, K., Ranabahu, A., 2008. Semantics enhanced services : METEOR-S, SAWSDL and SA-REST. *In IEEE Data Engineering Bulletins*, 31(3):8-12, 2008.

Domingue, J. et al, 2008. *IRS-III : A broker-based approach to semantic web services.* Journal of Web Semantics, 6(2) :109-132, 2008.

Dantan, J., Pollet, Y., Taibi, S., 2012. Semantic Indexation of Web Services for Collaborative Expert Activities, IADIS International Conference on Information Systems 2012, pp.57-64.

Pollet, Y., 2010. *Une approche algébrique pour la réutilisation et l'orchestration de services dans les systèmes d'information.* Revue Ingénierie des Systèmes d'Information. Journal Revue Ingénierie des Systèmes d'Information, vol. 15(5), pp. 63-88, 2010.

Levy, N., Losavio, F., Matteo, A., Ramdane-Cherif, A., Hadj Salem, H., 2011. Quality Standards Ontology for Web Service Discovery, *In XXXVII Conferencia Latinoamericana de Inforática (CLEI 2011)*, October 2011, pp.625-624, Quito, Ecuador.