STAR Lab Technical Report

From knowledge to interaction: from the Semantic to the Pragmatic Web

Peter Spyns & Robert Meersman
From knowledge to interaction: from the Semantic to the Pragmatic Web

ABSTRACT
Currently scientific attention with respect to the Semantic Web is mostly oriented towards syntax and semantics, at the detriment of pragmatics. This paper intends to draw the attention to the importance of “knowledge in context” (pragmatics) for ontologies-based applications. Thanks to its philosophical grounding, the VUB STAR Lab DOGMA ontology server is able to incorporate pragmatic knowledge, thus providing a powerful device to realise the vision of the Semantic, or rather, Pragmatic Web.

Categories and Subject Descriptors
H.1.1 [Systems and Information Theory]: general systems theory

General Terms
Design, Standardization, Theory.

Keywords
semantic web, ontology, database technology, computational linguistics, philosophy of language

INTRODUCTION
A lot of publications on the Semantic Web are about ontologies. Although there exist many definitions of ontologies in the scientific literature, some elements are common to these definitions: a computer ontology is said to be an “agreement about a shared, formal, explicit and partial account of a conceptualisation” [17,43]. In addition, we retain that an ontology contains the vocabulary (terms or labels) and the definition of the concepts and their relationships for a given domain. In many cases, the instances of the application (domain) are included in the ontology as well as domain rules (e.g. identity, mandatoriness, rigidity, etc.) that are implied by the intended meanings of the concepts. Domain rules restrict the semantics of concepts and conceptual relationships in a specific conceptualisation of a particular application domain. These rules must be satisfied by all applications that want to use – or “commit to” [16] an interpretation of – an ontology.

Authors who stress the dynamic aspects of the knowledge are mostly active in the field of intelligent software agent communication (e.g., [40]). They are concerned with adjusting the different conceptualisations of a particular domain by intelligent agents to one another in order to establish meaningful communication. Or stated in philosophical terms, how one software agent interprets the “possible world” of another agent. If the organisation of the world is completely different (semantics), or if the (formal) language used is incompatible (syntax), there will be no communication between agents. Even if these two latter issues pose no problems, then still communication could fail.

The situation in which the communication takes place can play a discriminating role for determining which concept is meant. Or the situation can provide additional meaning. These issues are to be situated not on the level of semantics, but rather on the level of pragmatics. As a consequence, we state, in agreement with Singh [37:p.82] and Kim [26], that the vision of the Semantic Web [3] will be realised as a Pragmatic Web.

As an illustration of the thesis put forward above, the term “pragmatic” must be disambiguated (i.e. linked to the concept appropriate for the current situation). In this paper, it is not a synonym of “practical, realistic, no-nonsense, down-to-earth”, which is the common interpretation of the word. Another, philosophical meaning, which is not to be retained either, originates from the writings of C.S. Peirce for whom pragmatics has to do with the purpose of information, its use and its evolution or change during use [7:p.237] 2. The linguistic acception of “pragmatics” is the one that is retained here, and will be explained below.

In the following section 0, the term “pragmatics” will be defined from a linguistic point of view and more details will be given on the relationship between semantics and pragmatics. Section 0 will describe a reference architecture for an ontology server oriented towards scalability and reusability [24]. It will be explained how pragmatic knowledge can be incorporated in this architecture (section 0) and how intelligent autonomous software agents would use this ontology server environment to achieve interoperability (section 0). Before a conclusion (section 0) is provided, section 0 gives on overview of related work.

SYNTAX - SEMANTICS - PRAGMATICS
In the fields of philosophy of language and (computational) linguistics, the triplet syntax – semantics – pragmatics as defined by Morris [34], is well known. He stated that syntax deals with the formal aspects of language, semantics with its meaning and pragmatics encompasses the entire communicative situation. In our view, this definition also holds for communication between software agents. Pragmatics not only includes non-verbal communication, but also the organisation of a discourse, i.e. the articulation of a written text or the history of a conversation. The discourse defines the context in which meaning is to be used or applied. Also, the sender and receiver of a message – see [46] – are part of the communicative situation. Pragmatics can specify or modify semantic knowledge. The above paragraph on how the term “pragmatics” should be interpreted clearly illustrates that a communicative situation allows to determine more accurately meaning defined out of context (= the three dictionary definitions (or concepts) for the term).

We use an example cited by Halpin [20:p.17] to illustrate our ideas. It concerns people’s age: in Japan a new-born is considered to be one year old instead of zero years – a different conceptualisation of counting age. In terms of formal languages one could define the concepts “age” related to a “person” by means of some property “range” (or by a role) over natural numbers. But how to express that when it concerns Japanese (but also Chinese, so let’s call it Asian) users the range starts from 1

---

1 In its general non technical meaning.
2 Note that this philosophical acception is related to the linguistic one, but the proposal of de Moor et al. [7] is, in our opinion, currently out of reach for software agents.
(instead of 0) ? Adding to the ontology a concept "AgeAsian" with specific property restrictions means that in e.g., DAML+OIL
3 there will be two files with instances (each with a different ontology header part). Note that Protégé can solve this more
elegantly by using facets when linking slots to classes [13].
Whenever a query is done for the age, the Western representation
should be transformed to the Asian one (and vice versa) – e.g., by
XSLT [11:p.24]. The direction and necessity of such a translation
would, again, be determined by the sender and receiver
participating in the communicative situation – e.g., query from an
"Asian-like thinking" user (be it human or agent). The loss of an
Ariane space rocket some time ago illustrates the consequence of
neglecting these issues (measures expressed in miles instead of
kilometres or vice versa).

Another example, also cited by Halpin [29:p.90], shows how
the pragmatic level can even influence the semantic level by
modifying the semantic specifications of a concept or relationship.
The example concerns two identical conceptual models (on
linking telephone numbers to persons), except for one having an
extra uniqueness constraint. In a hospital, each person being a
patient has only one telephone number (e.g., one patient per room,
one telephone per room), while in other situations a person may
be associated with several phone numbers (e.g., home, office,
mobile). The specificity of the pragmatics adds the notion of
uniqueness 4 to the meaning of the
"AssociatedTelephoneNumber" relationship. Again, DAML+OIL
needs two different files to represent these two "world views".

A third and final example is about the unique identification
of a person. Depending on the application, a combination of the
name and birthday might be sufficient, while another application
requires a unique identification number. Note that in this case,
pragmatics basically concerns the choice of the appropriate way
of identification.

Developers of ontology-based applications should take these
pragmatic issues into account. In the same way as a hand-shaking
protocol exists for communication networks, some "mind-
shaking" procedure will have to be developed for web services
and autonomous agent applications. Such a procedure could
consist of a stage during which the formal language for
information exchange should be determined (syntax), followed by
a synchronisation of the meaning of the concepts (semantics) on
basis of the particular context – e.g., purpose, time, date, profile,
… (pragmatics). We don’t go so far to assume that so-called
intelligent or autonomous agents are already intelligent or
autonomous enough that they are able to communicate with
another random agent on a potentially random subject (i.e. in a so-
called open environment). Such an agent would dispose of
capabilities including but not limited to on the fly ontology
aligning or merging, ontology language detection and
transformation etc.

Note that the pragmatic level is already in an approximate
manner included in many search engines and e-business
applications under the form of a user profile. A user profile
creates an interaction history and/or allows storing preferences to
restrict the scope of queries (for product or information retrieval
in general). Queries are thus contextualised and can provide more
relevant results for the user [19]. Ideally, a user profile is
expressed by means of concept labels instead of linguistic terms –
e.g., see [6].

THE DOGMA APPROACH
In this section, an overview of the DOGMA (Developing
Ontology-Guided Mediation for Agents) initiative [25], a VUB
STAR Lab research project, on ontology engineering is provided.
In a first section, theoretical and practical aspects are explained on
how to model, represent, and store knowledge in the DOGMA
framework. Then we’ll briefly sketch how knowledge is
accumulated and finally how knowledge is used and applied.

MODELLING, REPRESENTING AND STORING KNOWLEDGE
The DOGMA initiative is based on philosophical insights of
Wittgenstein [48] and Tarski [42]. In his search to define a
correspondence between reality and language, Wittgenstein in his
early period advocated that an atomic fact in the world
corresponds to an atomic proposition. Complex structures in the
world are reduced to (and reflected by) logical combinations of
elementary propositions 5. Tarski refocused the attention of
language philosophers by developing a meta-language to define
the semantics of a given language in a formally correct way.

We are aware that subsequent philosophers argued that
formal languages in order to be "applied" to reality still need
natural language (with its prescientific and preconceived
interpretation of the world) as the ultimate meta-language.
Nevertheless, we are convinced that the ideas of Wittgenstein and
Tarski can be validly applied to formal computer ontology
engineering. One of the goals is to achieve interoperability
amongst autonomous software agents that, by definition, are
formal constructs. Hiz supports this view by stating that "a
semantics of formal languages, a formal model, accepts the
fundamental similarity between phenomena in reality and the
elements of the formal language" [23:p.439], opposed to a natural
language.

In the style of Genesereth and Nilson [14], we thus define
formal ontologies in a logic sense, i.e. as "representationless"
mathematical objects that form the range of a classical
interpretation mapping from a first order language (assumed to
lexically represent an application), to a set of possible
("plausible") conceptualisations of the real world domain. We
introduce the double articulation of an ontology [39] by
decomposing it formally into an ontology base and into instances
of their explicit ontological commitments. The latter become
refined in our architecture as a separate mediating layer called
commitment layer. This also leads to methodological approaches
that naturally extend database modelling theory [31] and practice
[32], and so may in turn lead to scalable and reusable solutions
for ontology-based systems [25]. From the following paragraphs,
it will become clear that the first articulation (the ontology base)
corresponds to Wittgenstein’s atomic propositions, while the
second articulation (the commitment layer) matches Tarski’s
formal definition of semantics.

5 This position – known as the picture theory – turned out to be inadequate for daily language use, but seems to hold for formal languages. Note that Wittgenstein did not admit a meta-language.

3 www.daml.org
4 Similar ones are cardinality, mandatory, identity.
An ontology base consists of sets of intuitively plausible conceptualisations of a real world domain where each is a set of context-specific binary facts, called lexons, written as $\text{lex}_p \triangleright \text{term}_f, \text{role}, \text{term}_r$. Informally we say that a lexon is a fact that may hold for some application, expressing in that case that within the context $\gamma$ the term, may plausibly have term, occur in role with it. Lexons are independent of specific applications and should cover relatively broad domains. Lexons in a DOGMA ontology base are always "true", i.e. free of further interpretation.

To accommodate alternative models of reality, or even different versions, the ontology base may contain many different conceptualisations, even about the "same" real world domain, organised in so-called contexts – see the $\text{ist}(C,p)$ predicate of McCarthy [30] expressing that the proposition $p$ is true in context $C$. For the time being, we will consider contexts described by an identifier $\gamma$ having the generic property of disambiguating the terms, i.e. the combination $\text{lex}_p \triangleright \text{term}_f, \text{role}, \text{term}_r$ always will identify a concept (per definition unique) within that context. Possibly contexts can be constructed as expressions from others. Concept types are assumed to stand for the "referent" in the state of affairs independently of any lexical representation, attributes or properties.

The layer of ontological commitments mediates between the ontology base and its applications. The commitment layer is organised as a set of ontological commitments [18], each being an explicit instance of an (intensional) first-order interpretation of a task in terms of the ontology base. Each commitment is a consistent set of rules (or axioms) in a given syntax that adds specific semantics to the lexons of the ontology base. E.g., combinations of lexons with cardinality constraints. Commitments have a varying degree of generality. One can imagine a generic commitment that defines a particular but generic version of the "isA" relationship, while commitments related to the telephone example mentioned above are rather application specific. Commitments can also include derivation and inference rules – remember the derivation/transformation of Western into Asian age. Sets of ontological commitments can be regarded as reusable knowledge components. Such components are connected since they share the same ontology base.

ACCUMULATING KNOWLEDGE

For applications to effectively use ontologies, lots of data must be "ontologised". Different sources are to be used in order to gather data to be entered into the ontology server. A classical approach is to solicitate experts (see e.g., [21]). As an alternative to this time-consuming activity, various ways are investigated to have this activity done in a semi-automated way (we believe that the human will always remain in the loop as a validator and guarantee for quality). One can start from existing database schemas (mining databases – e.g. [9]), from textual descriptions (mining text corpora possibly combined with information extraction – e.g. [28]), from already existing ontologies that one wants to include (ontology merging – e.g. [29]) or consult (ontology aligning – e.g. [41]) and from (semantically annotated) data rich web pages – e.g. [51]. Once an ontology has been modelled, instances (as real data) are to be stored in an instance base. As these topics do not constitute the focus of this paper, we don’t give more details here.

USING AND APPLYING KNOWLEDGE

Applications operating in a particular communicative situation, e.g., banking or library agents, will want to use the knowledge base (ontology and stored instances): they define their interpretation(s) of the ontology, select the ontological commitments that satisfy their interpretation(s), and commit to them. I.e., programs/applications use ontological notions, and more importantly their corresponding instances, in the way they have been defined in the ontology base and further constrained in the commitment layer.

As the ontology base and the commitment layer are implemented with database technology, issues of scalability and reusability [33], robustness, security, etc. should be coped with. Object Role Modelling (ORM) [20], being a semantically rich modelling language has been selected as the basis for an ontology language that is to be extended within the DOGMA approach. This should not be a surprise as ORM, and its predecessor NIAM [45], decompose the universe of discourse of a database application into elementary natural language sentences. Another phase in the modelling process is to formally define the semantics of the elementary propositions (in terms of uniqueness, exclusiveness, mandatoriness, ...). The ORM modelling methodology is currently being adopted for ontological engineering. As native commitment language we similarly develop O-RIDL as an ontological extension of the RIDL language (e.g. [45]).

DOGMA & PRAGMATICS

In this section, it will be shown how the DOGMA architecture takes the pragmatic level into account. More in particular, there is a relationship between the contexts of the ontology base and the organisation of the commitments into sets. From the previous section, it becomes clear that the commitment layer is the place where pragmatics are taken into account. For each application (or even task), there exist a corresponding set of commitments (that we call pragmatic view) to the ontology. As commitments can be grouped together, one can organise them in such a way that a pragmatic view actually reflects a particular communicative situation.

Contexts in the DOGMA lexon structure appear as abstract (in principle even semantics-less) identifiers that provide an ability for an internal organisation of an ontology base. Naturally—or involuntarily—"meaning", or at least an interpretation, is assigned to a context as seen by an ontology author’s interpretation. Sowa notes that a context is a package on information about a separated chunk of the world [38,p.184]. The DOGMA contexts exactly fulfil the same function, i.e., organising the lexons into meaningful groups. We support the position that knowledge modelling and representation always reflect the view

---

6 In our opinion, a purpose or task independent ontology can never be achieved (think e.g. of the granularity of the concepts – also see [38,p.171]). But thanks to a collaborative modelling process, an ontology should attain a high degree of application independence but will somehow remain limited to a family of related applications.

7 Their truthfunctionality is guaranteed by the human modeler who acts a “deus ex machina”.

8 A set of generic, i.e. only used by other views, commitments could be called a semantic view.
of the modellers on the domain to be modelled, although the ideal should remain to be as independent as possible of any purpose or application – see footnote 6. Also this position is backed up by Wittgenstein, albeit in his later period, who stated that absolute elementary propositions don’t exist, but form part of a system – we would call it a communicative situation. Thus, contexts reflect in part the organisational view of the modellers on the domain concerned. From a declarative semantics point of view, the part(s) of contexts in an ontology base committed to an application via pragmatic views may be seen as one particular (extensional) possible-worlds model [38] for the logical theory defining that application.

Commitment rules can be easily grouped in terms of DOGMA contexts in the ontology base. As a result, an application can select lexons (from the ontology base) with specifically defined semantics (in the commitment layer) based on pragmatic knowledge, e.g., the purpose, time, date, task, profile, ... As Sowa puts it: "some agent for some purpose must pick and choose what is relevant [38:p.187]". However in the DOGMA approach, no formal mechanism enforces a correspondence between a context and commitment. An ontology author does not necessarily define contexts in perfect accordance with the pragmatic views needed by an application developer. In practice, application developers should be involved in the process of collaborative ontology modelling in order to obtain a widely shared agreement on the ontology (base). As a side-effect, contexts will mirror better (potential) pragmatic views.

Similar applications, i.e. with the same pragmatics, can reuse/inherit (a set of) commitments from each other. On the one hand, this facilitates new applications to commit to and use the ontology, and on the other hand, successful “pragmatically” established commitments will likely become “popular” and therefore a de facto trusted resource in their own right. One can even think of a layered structure of semantic and pragmatic views – see note 8. We conjecture that, in general, a context will cover a broader “chunk of knowledge” than a pragmatic view as the former helps to organise knowledge of a family of applications (shared conceptualisation about a domain) while the latter in essence concerns one application (and its internal data model) designed for a specific purpose. However as knowledge modelling is still very much an art, no rigid rules exist on what the “span” of a suitable context, the characteristics of a reusable pragmatic view, and a good measure of the degree of their correspondence could be. Practice should show. Of course, this implies that the various commitments as well as the contexts are clearly and unambiguously described – basically what the purpose and rationale are for organising knowledge (commitment rules viz. lexons) in a certain manner.

Ontology servers can advertise their semantic and pragmatic knowledge, i.e. an ontology (subdivided in contexts) and its related commitments (grouped into semantic and pragmatic views). Autonomous agents are able to select a pragmatic view (that corresponds to a context in the ontology base). In our view, a human will always have to select an identifier of a pragmatic view and incorporate it in an agent 11. Note that an application may commit to more than one pragmatic view as long as they are consistent one with another. The meaning in such case is that all the pragmatic views together form one complete interpretation for that application [16]. This complies with Kripke’s definition of possible worlds [27].

**DOGMA & AGENT COMMUNICATION**

In its minimalist set-up, 12, one can picture two intelligent software agents (e.g. one looking for a service and one offering a service) sharing an agreement on the purpose and context of their respective tasks expressed by a formal conceptualisation stored in a DOGMA ontology server. We refer to the notion of possible worlds as already mentioned in section 0. A number of formalisms exists that describe possible worlds semantics, the most well known from Kripke [27] and Hintikka [22]. We retain their modal operators (possibility and necessity) and translate it into the DOGMA “parlance”. Possibility (= to be true in at least one possible world) translates into the choice of a particular selection of lexons by an agent (e.g. looking for a specific service) that intends to use an application (e.g., a web service) via a pragmatic view (that groups commitment rules). The notion of necessity (= to be true in every possible world) translates into a requirement for a lexon to be part of any valid selection within a context for that application.

Establishing meaningful communication between the agents in a minimalist setting boils down to exchanging the appropriate pragmatic view identifiers. As stated in [41:p.47], "interoperation between agents is obtained by the agent’s commitment to use a shared vocabulary and to associate a common meaning with the terms of the vocabulary". When referring to DOGMA, it means that agents have to have at least one commitment rule in common. Otherwise, they have no "common ground" since a commitment defines (or rather restricts) the semantics of lexons. Or stated otherwise, the worlds of these agents would not intersect. One could imagine some internal decision logic (e.g. based on some combination of time and purpose dependent factors) that allows an agent to choose the most appropriate commitment (so-called deliberative agents – [41:p.50]). When referring again to the "age example", one agent would have to choose the appropriate pragmatic view (asian vs. western) and take care of data transformation (recalculating the age) if necessary. One can think of a search engine disposing of a user profile that is used to choose the appropriate commitment.

Just as the data has been taken out of software applications and stored in a (relational) database, it becomes evident to take the definition of the agent’s conceptualisation of its own possible world (= its external “beliefs”) out of the autonomous software agent’s implementation and store it in a central repository (= ontology server).

---

9 He advocates that space and time are not discriminating enough to organise a meaningful situation.
10 Therefore, the use of a single label for a context is most probably insufficient – see [15] – but will do on a temporary basis. More research on this topic is to be done – e.g. [4].
11 We don't want to end up in an infinite circle in the sense that the pragmatic knowledge would be described by means of a formal language of which over time various variants would be created, which would then have to be aligned, transformed etc. to allow interoperability. A human developer has to synchronise the internal data model of the agent with the pragmatic view(s) an agent should select.
12 Making abstraction of a.o., aligning and merging of conceptualisations.
Note that we do not believe that an agent will ever become so intelligent that it could and transform the formal representation of its own ontology, and align its conceptualisation using its pragmatic knowledge. Agents are doomed to stay on the symbolic level and cannot refer to the state-of-affairs or universe of discourse. Only humans can ground representations by linking them to “the real thing” in the world (“reference” – see [12]). They ensure the correspondence between the agent conceptualisation of its world and the actual state of affairs.

RELATED WORK
As already mentioned before, the idea of introducing pragmatics for the Semantic Web has also been put forward by Singh [37] and Kim [26], albeit from a different point of view. Nor do they present technical solutions.

Contexts have been and are the subject of occasionally intense study in their own right, especially in AI; for example [30] or [38]. Philosophers like Kripke [27], Hintikka [22] and Dunn [10] have studied related topics as possible worlds, model sets and their relationships.

Relevant to model the behaviour of autonomous software agents is the architectural triple “beliefs, desires and intentions” [1], as used for instance in dialogue applications [44].

Several researchers try to achieve semantic interoperability of software agents by transforming one description logics language into another one – e.g. [11] – or by aligning the ontologies of the agents – e.g. [40] presents a method to approximating concepts when aligning. Patzian and Vindigni [35] suggest to use natural language terms related to concepts to achieve an agreement on a shared representation of a domain. The context in which the terms are used helps to select the correct concept.

One of the intended uses of the DOGMA ontology server is to function as a mediator. As such, the topics described in this paper blend in with the research on research on data mediators in general – e.g. see [2,47]. An overview of ontology servers in general is given in [8:p.216]. Mediator systems are presented in [36].

We want to draw the reader’s attention to the parallel that exists with the database paradigm. One can think of the shared and formal conceptualisation of a domain as the conceptual database model, the lexons and contexts together with the commitment rules as the logical model, and the semantic and pragmatic views as the external model. Note that this double articulation copes more elegantly than traditional ontology languages with the issues as illustrated in section 0 – see also [39]. Nevertheless, we prefer to use the linguistic paradigm as it captures better the dynamic aspects of a communication – e.g. of software agents. But for the practical realisation of all this, we rely on solid database technology [25].

CONCLUSION
We have presented the notion of pragmatics and shown its importance for the Semantic Web. The DOGMA architecture easily allows the incorporation of a pragmatic knowledge level, thanks to its specific representation of lexons in an ontology base on the one hand and the separation of application related issues in a commitment layer on the other (= its double articulation). As argued elsewhere [25], the separation also greatly favours reusability and scalability. As a result, we believe that the DOGMA approach and architecture, by virtue of the underlying database technology and methodology, are well suited for building complex and large-scale applications, e.g. the DOGMA ontology server, that materialise the vision of the Semantic, or rather, Pragmatic Web. Unavoidably, the next step is to implement an agent prototype that connects to the DOGMA server via pragmatic views.

ACKNOWLEDGMENTS
The authors are grateful to the other STAR Lab members for their support and stimulating discussions, in particular Mustafa Jarrar.

REFERENCES


