A Devolved Ontology Model for the Pragmatic Web

Iain Duncan Stalker and Nikolay Mehandjiev
School of Informatics,
University of Manchester,
PO Box 88, Manchester,
M60 1QD, UK
iain.stalker@manchester.ac.uk

Abstract: Devolved ontology is an approach to ontology modelling and (co-) evolution which was developed in connection with agile partnerships. Inviting parallels between agile partnerships and the context of the Pragmatic Web suggest that this has potential value in realising the vision of the Pragmatic Web [SMD06]. This is especially clear in their respective uses of ontologies and in particular the demands made of supporting structures. We motivate and introduce the devolved ontology model and show how to use this to promote semantic alignment and thereby support communication.

1 Introduction

The "Pragmatic Web" is a complementary trend emerging in the management of distributed, heterogeneous information and resources which identifies and aims to ease or circumvent some fundamental problems perceived in the Semantic Web [SMD06]. Proponents of the Pragmatic Web centre upon the difficulties arising from the particular use of ontologies in approaches to the Semantic Web.

> *The vision of the Pragmatic Web is ... to augment human collaboration effectively by appropriate technologies, such as systems for ontology negotiations, for ontology-based business interactions, and for pragmatic ontology-building efforts in communities of practice. In this view, the Pragmatic Web complements the Semantic Web by improving the quality and legitimacy of collaborative, goal-oriented discourses in communities* [SMD06].

Our interest in ontologies derives from our work in supporting agile partnerships, where partners combine their respective strengths opportunistically to improve competitiveness; and typically form with a target project in mind, i.e., are goal-oriented. We are particularly interested in using ontologies to represent knowledge and information in dynamic, evolving domains in which discourse makes use of concepts from multiple (application) contexts. We find strong, inviting parallels between the motivation for our work and the interests of Pragmatic Web, especially as captured in the following quotation.

> *Ontologies are not fixed, but co-evolve with their communities of use. Communication partners have to agree continuously on what they can*

assume to be the shared background. This is especially important in an organizational context where parties from different professional, social, and cultural backgrounds need to understand each other. In order to enable the use of the Web for communicating, agreeing upon, and co-operatively modifying ontologies, the support provided by the Semantic Web is insufficient. An ontology is an agreed-upon conceptual specification used for making ontological commitments. The crucial question is: how do human agents commit and renegotiate their meaning commitments? And what kind of socio-technical infrastructure is required to leverage those conversations? [SMD06].

Agile partnerships are dynamic, open networks of entities which assemble opportunistically to fulfill a particular purpose; examples include Virtual Enterprises, Supply Chain Networks and eMarketplaces. Enabling software technologies for such partnerships must both capture the distribution of intelligence or expertise and facilitate meaningful communication. Multiagent systems offer much to foster the open nature of agile partnerships. For example, *ad hoc* interaction with new arrivals is supported through *agent communication languages* and *interaction protocols* [Fe99, Wo02]. Nevertheless, there are limitations: communication in multiagent systems presupposes a common ontology, which is typically fixed in both content and semantics. Yet, the nature of agile partnerships suggests neither a fixed ontology nor a unique semantics is appropriate. In agile partnerships ontologies do indeed "co-evolve with their communities of use". We have developed an approach which supports this, which we see as a potentially useful contribution to the vision of the Pragmatic Web. In this paper we make a clear distinction between the use of ontologies in formalising a domain of interest and their use in supporting communication. We use this to motivate our approach. We also indicate how a combination of a devolved ontology model, agents, utility functions and interaction protocols provides for negotiated concept evolution which extends to open environments generally.

2 Preliminaries

In this section we present the formal apparatus used to implement the particular devolved ontology model we introduce in Section 5. Specifically, we briefly introduce Formal Concept Analysis, Partially Shared Views and we recall some relevant aspects of the Theory of Utility. We assume a familiarity with some aspects of order theory (we recommend [DP02]) and with multiagent systems [Wo02, Fe99], though this is more intuitive.

2.1 Formal Concept Analysis

Formal Concept Analysis (FCA) [GW99] is a powerful, elegant method of analysis which identifies (conceptual) structures within data sets. The qualifier *formal* emphasises that these are mathematical notions, which do not necessarily capture the

	Size			Distance from Sun		Moon	
	Small	Medium	Large	Near	Far	Yes	No
Mercury	X			X			X
Venus	X			X			X
Earth	X			X		X	
Mars	X			X		X	
Jupiter			X		X	X	
Saturn			X		X	X	
Uranus		X			X	X	
Neptune		X			X	X	
Pluto	X				X	X	

Table 1: A Simple Context for the Planets; after [DP02].

everyday use of the terms. We dispense with the qualifier for convenience.

Definition 1 (Context and Concept) *A context is a triple (G,M,I) where G and M are sets and $I \subseteq G \times M$. G is the set of* objects*, M is the set of* attributes *and I is an* incidence relation*. We write gIm for $(g,m) \in I$.*
Let $A \subseteq G$ and $B \subseteq M$. Define $A^{\triangleright} = \{m \in M \mid gIm, \forall g \in A\}$, then $A^{\triangleright}$ is the set of attributes shared by all objects in the set A. Similarly define $B^{\triangleleft} = \{g \in G \mid gIm, \forall m \in B\}$, then $B^{\triangleleft}$ is the set of all objects possessing the attributes in the set B. These maps are called derivation operators*. A* concept *of the context (G,M,I) is a pair (A,B), such that $A^{\triangleright} = B$ and $A = B^{\triangleleft}$. The* extent *of the concept (A,B) is A and the* intent *is B.*

Definition 2 (Concept Lattice) *Denote the set of all concepts of a context $\mathscr{B}(G,M,I)$, or simply $\mathscr{B}$ where the context is clear. Define a partial order, $\leq$, on $\mathscr{B}$ as follows: $(A_1,B_1) \leq (A_2,B_2) \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow B_1 \supseteq B_2$. Then $(\mathscr{B},\leq)$ is called the* associated complete lattice of concepts*, or simply* concept lattice*, of the context (G,M,I).*

We illustrate the basics of FCA through a simple example. Table 1 illustrates a simple context for the planets (objects) of the solar system, categorising these according to a number of attributes such as size, distance from the Sun and whether a planet has a moon. Consider $\{$Mercury, Venus$\}^{\triangleright} = \{$size-small, distance-near, moon-no$\}$. and $\{$size-small, distance-near, moon-no$\}^{\triangleleft} = \{$Mercury, Venus$\}$. Thus, $(\{$Mercury, Venus$\}, \{$size-small, distance-near, moon-no$\})$ is a concept of the simple context of Table 1. Similarly, $(\{$Mercury, Venus, Earth, Mars$\}, \{$size-small, distance-near$\})$ is a concept of the simple context of Table 1. Moreover, since $(\{$Mercury, Venus$\}, \{$size-small, distance-near, moon-no$\}) \leq (\{$Mercury, Venus, Earth, Mars$\}, \{$size-small, distance-near$\})$ the former is a subconcept of the latter.

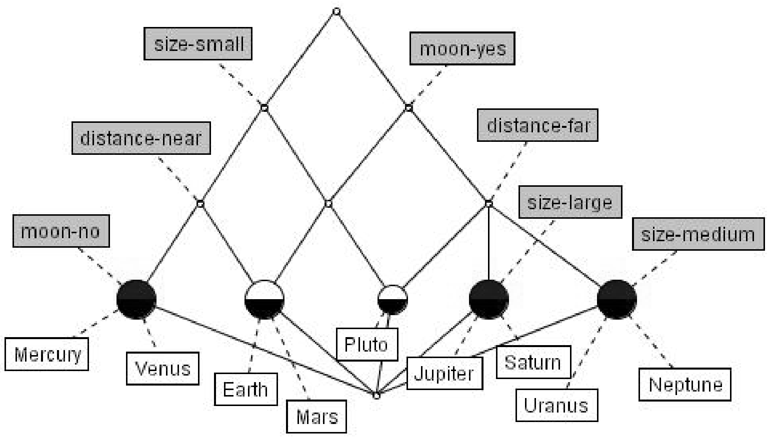

Figure 1: A Concept Lattice for the Planets from Table 1; after [DP02]. The concept lattice is read in the following way: objects accumulate from the bottom upwards; and attributes accumulate from the top downwards. For example, the concept at the node marked *dn* includes *dn, ss* as attributes and Me, V, E, Ma as objects; the concept at the node marked E Ma includes *ss, dn, my* as attributes and E Ma as objects.

We can provide pictorial representation of the concepts of our context and their interrelations using a Hasse diagram [DP02]; see Fig. 1 [1]. The concept lattice for a given context provides a direct manner in which to identify whether a relationship exists between two given concepts; and further, clarifies the nature of this relationship.

2.2 Partially Shared Views

Partially Shared Views (PSV) is a scheme to facilitate communication among disparate groups [LM90]. It arose in the context of template-based office communication systems. Central to the scheme is a number of semistructured templates for different types of objects. The term *type* is used to denote a particular class of objects and this notion corresponds directly to the notion of *concept* in an ontology. Five cases of communication are presented, ranging from no common language to a coincident common language, where *language* is used in a restricted sense to denote a set of (object) types. Of particular interest to our application is the fourth case (*Internal Common Language*, [LM90] p.13). Here a common language is included in each of the group languages. A type hierarchy identifies a partially ordered set of types: typically, ordered according to attributes or properties. The idea is to find

[1] The node colourings provide useful information concerning filters and ideals [GW99] furnished by the tool used to produce this figure, *Concept Explorer* (`http://sourceforge.net/projects/conexp`). This information is additional to our current purposes, thus we do not discuss here.

a common superconcept to support communication between agents using different specialisations of the common ontology. Suppose *Agent A* uses ontology *Application Ontology A* to construct a message, `message`, which contains for convenience a single concept, and sends this to *Agent B*. Further suppose that *Agent B* uses *Application Ontology B* and that each application ontology extends a common ontology, but is otherwise different. Notwithstanding available translation rules, there are two cases to consider. If the concept in `message` *is* contained in the common ontology, then *Agent B* understands the communication. Otherwise, the concept in `message` *is not* contained in the common ontology, and *Agent B* will not understand the communication. In this case, the concept is mapped to an appropriate superconcept in the common ontology: this used in its place; *Agent B* understands the revised communication, which we note is strictly an approximation of the initial message. Common interapplication terms means that the mapping to a superconcept need not always go to the common ontology.

2.3 Theory of Utility

Formally, a *Utility Function* (for an agent $\mathscr{A}$), $u_{\mathscr{A}}$, associates with each possibility from a set of outcomes or states, $\omega \in \Omega$, a measure — a real number $u_{\mathscr{A}}(\omega) \in \mathbb{R}$ — to reflect the "enjoyment" which would be derived from each state. That is, $u_{\mathscr{A}} : \Omega \to \mathbb{R}$. Utility functions offer a means to model the *preferences* of an agent and as such have attracted much attention from researchers in (computational) multiagent systems [Wo02], especially when addressing issues of negotiation.
The *Theory of Utility*, which informs utility functions, developed as a pillar for *Game Theory*, but in the modern sense it stands apart and finds application in other contexts, including economics and decision theory [LR57]. These other contexts often provide useful additional distinctions and techniques. Of particular note is the notion of a *focal point* [Su95]: informally, this is some salient feature which provides a focus *"for each person's expectation of what the other expects him to expect to be expected to do"* [Sc60]. In the sequel, it is the *notion* of a focal point as captured in the above quotation which interests us, rather than the development of the *Theory of Focal Points* [Su95]. In particular, we consider the minimum information which must be communicated by agent to 'get its message across' to be a focal point. We enlarge upon this in Section 5.

3 Ontology

Informally, an ontology comprises of a set of concepts and a set of relations which describe and constrain how the concepts refer, interrelate and combine. Recent interest in ontologies has led to a number of definitions of the term "ontology", see e.g. [NK04] or [GFC04], but we prefer that offered by Guarino: an explicit, partial account of a conceptualisation, where a conceptualisation identifies "a set of informal rules that constrain the structure of a piece of reality, which an agent uses in order to isolate and organize relevant objects and relevant relations" [GG95].

The value—in terms of reusability and portability—of a conceptualisation and *a fortiori* an ontology derives in part from its dependence on a given viewpoint. Informally, a viewpoint signifies the position taken by some agent when considering some "piece of reality" or domain of interest; and accommodates, *inter alia*, any perceptual, societal, environmental, linguistic, technological and cognitive constraints which appertain, including the intended use of that knowledge. An ontology deriving from a shared conceptualisation is likely to be more generic, perhaps more widely applicable and thus more valuable. We consider this prime motivation for a negotiated formalisation, especially for a domain of discourse.

In Computer Science, ontologies are typically used for one of two purposes: to formalise a domain of interest; or to support communication through a controlled, unambiguous vocabulary. While it is possible and often instructive to view the second as a special case of the first—in that we formalise a domain of discourse—their respective, underlying motives are fundamentally different.

1. *Formalising a Domain.* This is an exercise in (knowledge) engineering. We build an abstract model or construct a theory which ideally gives a precise and accurate account of the salient aspects of a domain of interest; which can be substantiated by practice or experiment. Thus, *objectivity*, i.e., independence of the account from the observer, is of primary importance. Typically, defining a substantive concept within a given domain involves agreement at two levels: we must identify what objects exist in our (shared) conceptualisation; and how these objects are characterised.

   Implicit in our theory is an ontological commitment: by describing some phenomenon through the use of denoting symbols, we are committed to the existence of certain entities and relations among these. This echoes perhaps the most familiar theory of ontological commitment; that of Quine, which claims in essence that one is committed to an entity if one refers to it directly or indirectly; cf. [Qu48].

2. *Supporting Communication.* Supporting communication is an exercise in pragmatics. *Pragmatics* is a subfield of linguistics which investigates the nature of communication in concrete situations. In particular, it distinguishes two intents within a given communicative act—usually verbal, but these apply in a wider sense—namely [Le83, SW86]: *informative intent* or the (interpretive or referential) meaning of the sentence; and *communicative intent* of the intended meaning of the speaker. Of especial interest in supporting communication are the so-called *deictic* aspects, which, in a general sense, confirm that valid interpretation demands knowledge of the context in which the communication occurs. This suggests that we must assume the viewpoint of the agent responsible for a given communicative act to receive the communicative intent for

the specific, concrete situation; and conversely, that, to ensure that the communicative intent is conveyed, a communicating agent should not presuppose that its viewpoint prevails in a domain of discourse.

The nature of ontological commitment in supporting communication differs markedly from that arising when formalising a domain: fewer concepts and relations are necessary; and importantly less structure is required.

In our opinion, the failure to maintain this dichotomy is one, significant cause of the delay in delivering on the promise of ontologies for communication; and frustrates much of the interaction between those active in the two different aspects. This is particularly evident when, as a first step to communication among partners from different domains, ontological alignment is sought in a manner which is tantamount to formalising the domain of discourse. There is a perceived need to agree on precise concept definitions and much is made of the merging of ontologies to achieve this. Accordingly, independently of method, agreement is sought at two levels: the identification of what objects exist in the (shared) conceptualisation; and how these objects are structurally defined. Yet, for a given domain of discourse, we—as individuals acting upon the world—are capable of entertaining simultaneously a number of conceptualisations which may be inconsistent, even contradictory or at different levels of granularity. We choose the most appropriate to the task at hand: we select according to context. As such, it is not convenient nor desirable to fix a unique characterisation of the domain of discourse. Indeed, such a choice often proves to be an impediment. Thus, in a practical sense maintaining the dichotomy means that we treat communication as a *de facto* exchange of a minimal sets of *essential* tokens of information; and we do not impose our ontology onto the communication.

4 Our Approach: A Devolved Ontology Model

Informally, a *Devolved Ontology Model* comprises of a core ontology and a number of extensions of this into peripheral and interapplication domain ontologies. It is a structure to facilitate ontological and semantic alignment among communicating entities. The core ontology provides a common ground for understanding among partners and is central to the partnership. The concepts included within this are agreed through negotiation of all partners. As such, the responsibility for the evolution and maintenance of the core is shared by the partners. Each peripheral ontology represents an extension of the core ontology into an application domain. The responsibility for the evolution and maintenance of each peripheral ontology *devolves* upon the appropriate partner or partners. This includes the responsibility for extending the core into the particular context and ensuring that the peripheral ontology remains consistent with the core. Since two partners may share a number of concepts which are not part of the core, we recognise the existence of interapplication domains and ontologies. The responsibility for the initial extension of the core into the interapplication ontology devolves upon two agents jointly; for further extension into each

application ontology devolves onto the appropriate single agent.
Crucially, devolving responsibility upon the appropriate partner (respectively partners) includes leaving the choice of appropriate syntactic structure to it (respectively them). Therefore, the first step in creating a formal devolved ontology model is the removal of syntatic aspects: structures are initially flattened. We propose that a given concept has a number of tokens, e.g. a set of attributes, associated with it. The tokens used to represent the concept (at a particular instant) are selected according to context, projecting away from those which are redundant to leave only an essential subset. We refer to the full set of tokens as the global (domain) concept: this may include inconsistencies. In the special case where the tokens are the same for each participant, we call this a common (domain) concept. To compensate for the removal of syntactic structure, it is imperative that we find some "natural" structure and allow this to emerge. In the development of the model in Section 5, we use *Formal Concept Analysis* (FCA) [GW99], *Closure Operators* (see, for example, [DP02]), and *Lattice Theory* [Bi67] to capture these ideas; and to provide sufficient rigour for systematic treatment. The selection of tokens as needed and the appeal to a "natural" structure allow our ontologies to be minimal and *self-constructing.*

5 Concept Negotiation

In the absence of any other informing principle, we can assume that an agent is reluctant to alter its knowledge base unless there is some (positive) payoff. Moreover, once motivated to revise its knowledge base, it will seek to minimise the extent of any change. Accordingly, in any concept negotiation we have a natural focal point (cf. [Su95]) for each agent: namely, those *essential details* which must be conveyed to ensure that the transaction is *appropriately informative*. For the sender, this identifies a lower bound for the concept under negotiation and any acceptable alternatives reside on some chain connecting it with the original concept. For the receiver, the closer he gets to this, the better: of course, he may not know what this is, thus it is in his interests at each stage to strip away attributes.

5.1 Using FCA and PSV to Create a Devolved Ontology Model

In PSV a *view* is defined to be *"a set of object types and their relations. A view V_2 is subtype of V_1 is some of the message types in V_2 are specializations of ("children of") the message types in V_1"* (p.16) [LM90]. FCA provides use with an appropriate formalism through which to realise PSV.
Suppose that the structure of our domain (i.e. its ontology) is comprised of a common ontology and two (main) application ontologies, A and B. Each of the ontologies identified above is a proper subset of a (notional) global ontology which includes all concepts in the domain. Generalisation identifies a subontology relationship and so the common ontology is a proper subset of each of the application ontologies. In fact, the common ontology is a subcontext of the application ontologies [GW99].

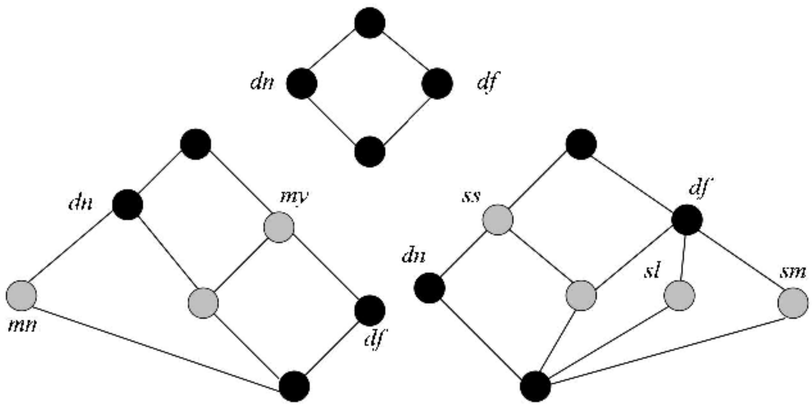

Figure 2: Common ontology (top), application ontology A (left) and application ontology B (right). In each application ontology, the common ontology indicated by black nodes. The common concept lattice is a factor lattice of each application domain lattice *cf.* [GW99]. Moreover, the application domain lattices are factor lattices of the global domain lattice of Fig. 1 and provide an *atlas decomposition* of this [GW99]. Our intention here is to illustrate structure, thus, for clarity in the diagram we omit the planets associated with each node.

We return to the context of Table 1. As our common ontology, we consider the categorisation the planets solely according to distance from the sun. As our application ontology A, we consider the categorisation the planets according to distance from the sun and the size. As our application ontology B, we consider the categorisation the planets according to distance from the sun and the presence of a satellite (moon). In each case, we enlarge the common ontology by (order-) embedding the *additional* concepts from each augmented ontology into the common concept lattice, as illustrated in Fig. 2. As such, the common ontology is a sublattice of each augmented ontology. Fig. 1 illustrates our (notional) global ontology and includes all of our concept lattices. The common ontology and each of the augmented ontologies is sublattice of the notional global ontology as illustrated in Fig. 2. Borrowing the term *view* from [LM90] we consider the above ontologies as (defining) views of the planets in our solar system. *Common View*, V_C: the planets categorised as near to the sun or far from the sun according to their positions inside of or outside of the asteriod belt, respectively. *Augmented View A*, V_A: an augmentation of the common view to include a consideration of planet size. *Augmented View B*, V_B: an augmentation of the common view to include a consideration of satellites (moons) *Global View*, V_G: an all-encompassing view which we associate with the domain as a whole: essentially, a superset of V_A, V_B and V_C.

Consider, *Agent A* sends a message, `message` = (... {size-medium, distance-far} ...), which *Agent B* does not understand, as the concept does not exist in its view,

V_B. Thus, if appropriate, we replace it with closest superconcept in the common view, V_C: from Fig. 1 we see is {distance-far}. We formalise the notion of closest superconcept using the partial order: let C be the concept of interest, $(\mathscr{B}_G, \leq)$ be our global concept lattice and let $(\mathscr{B}_C, \leq)$ be our common concept lattice. Let $C^u_G \subseteq \mathscr{B}_G$ denote the set of all of superconcepts of C in the global concept lattice $\mathscr{B}_G$ (the *upward closure*). The *closest superconcept* is C_T (T for target), where $C_T \in C^u_G \cap \mathscr{B}_C$ and $C_T \leq C', \forall C' \in C^u_G \cap \mathscr{B}_C$. Formally, C_T is a *minimal element* of $C^u_G \cap \mathscr{B}_C$. This is not necessarily unique, but we omit discussion of this here. Essentially, the mapping to the appropriate superconcept is a projection away from those attributes which are not in the common ontology.

5.2 Utility Functions

FCA provides a way to realise aspects of PSV and together these give rise to a particular instance of a devolved ontology model. While this is model, it provides merely the *what* of concept negotiation for a set of interacting entities, leaving us to determine through other methods *when* and *why* these should seek to negotiate. This is the role of utility functions. We need to equip our agents with these. We discuss this with simple examples in the current section. We assume that the decision to negotiate when faced with a novel concept depends, *inter alia*, on the *importance* of the third party(ies) involved; the *worth* of the (current) transaction; and the *cost-benefit* of admitting the concept. Moreover, the import of each of these depends on the stage of negotiation. For example, the cost-benefit of admitting the concept is unknown in the initial stages and has little impact on the decision to proceed with negotiation. When admitting the concept, the cost-benefit is a dominant factor. *Both* the receiver and sender can choose whether or not to enter into a negotiation over a novel concept. Thus, each could be equipped with a utility function. The decision to admit a novel concept belongs to the ontology agent associated with the ontology to which the novel concept would be admitted.

We take a simple approach. For each of *importance*, *worth* and *cost-benefit*: we identify a number of criteria; we allow the user to compare and rank these and we normalise the user rankings to provide a set of weights, $w_i \in [0,1]$, with $\sum_{i=1}^n w_i = 1$, where n is the number of criteria. For each utility function, we allow the user to set a threshold value, $u \in [0,1]$, which must be exceeded (for the utility measure to be worthwhile). Thus, we derive utility functions of the form: $U(U_i, U_w, U_c) = w_1 f_1(U_i) + w_2 f_2(U_w) + w_3 f_3(U_c)$, where U_i, U_w and U_c denote the utility (sub)functions for *importance*, *worth* and *cost-benefit*, respectively; the $w_i, i = 1,2,3$ denote weights; and $f_i, i = 1,2,3$ are functions (of the appropriate arguments) which return a value in [0,1]. Each of the utility (sub)functions takes a form similar to the total utility functions.

As a simple example, consider *cost-benefit*. Suppose we identify and rank the criteria as in Table 2. Normalising and then averaging [2] the entries of Table 2 leads to a set of

	Frequency f_r	Frequency f_t	Concepts N_A
Frequency (relative) (f_r)	1	5	3
Frequency (time) (f_t)	1/5	1	1/7
Auxiliary Concepts (N_A)	1/3	7	1

Table 2: Criteria for Cost-Benefit. Figure in cell ij indicates the relationship between criteria i and j as follows: 1 - indifferent; 3 - i is slightly more important; 5 - i is more important; 7 - i is significantly more important; 9 - i dominates. 2,4,6, and 8 are intermediates. ji is the reciprocal of ij.

average values which we use to construct a cost-benefit utility (sub)function to reflect our preferences: $U_c(f_r, f_t, N_A) = 0.59\,c_1(f_r) + 0.08\,c_2(f_t) + 0.33\,c_3(N_A)$, where, for simplicity, we might choose simple threshold functions for c_1, c_2 and c_3. Analogous procedures allow us to derive weights and attendant utility functions for *importance* and *worth*.

5.3 Negotiation Protocols

Having presented the *what*, the *why* and the *when*, it remains to show the *how*. Negotiation protocols provide this. The manner in which an agent responds when faced with a potential case for (concept) negotiation is informed by the nature of the relationship with the third party(ies) involved. This includes considerations of trust, vested interests, the degree of acquaintance, and so forth. The intangible nature of these often proves an impediment to the construction of satisfactory models[3]. We consider this information beyond the more immediate, objective measures captured in (our) utility functions and thus provide a choice of protocol through which to negotiate. For example, if one trusts implicitly the third party, then one might comfortably seek his opinion of the usefulness of a concept in future communications, secure in the knowledge that a fair response is obtained. The choice of protocol can be derived from an agent's list of acquaintances, cf. [Fe99], or from the values in the utility functions when deciding whether to negotiate, cf. Subsection 5.2, or a combination of these. We assume the utility functions discussed in Subsection 5.2. For simplicity, we also assume a single third party. We present an example protocol motivated by simplicity or progressive effort, which for convenience we call *Protocol A*.

[2] *Normalisation.* Let $e_1, \ldots, e_n$ denote the entries in a given column. The normalised entries are $\tilde{e}_1, \ldots, \tilde{e}_n$, where $\tilde{e}_j = e_j / \sum_{i=1}^{n} e_i$, $j = 1, \ldots, n$.
Averaging. Let $\tilde{e}_{k1}, \ldots, \tilde{e}_{km}$ denote the normalised values in row k. The average for row k is $\overline{e}_k = \sum_{i=1}^{m} \tilde{e}_{ki}$.

[3] Naturally, the same argument can be levelled at notions of importance and worth presented above, but we feel that a greater degree of objectivity obtains for these.

5.4 Protocol A: Simplicity

The protocol begins once two agents, *Sender* ($\mathscr{S}$) and *Receiver* ($\mathscr{R}$) (say), have agreed to negotiate over a concept which is not shared. We assume that responsibility for initiating the negotiate ontology protocol devolves upon $\mathscr{S}$ as it was its message which was not understood. This applies even when the message of the $\mathscr{S}$ is some form of response to $\mathscr{R}$. This is a debatable choice: convincing cases can be made for other options. $\mathscr{S}$ proposes negotiation to $\mathscr{R}$ prior to contacting its own ontology agent (and other interested parties) because *inter alia* there is no guarantee that $\mathscr{R}$ will participate in any negotiation; $\mathscr{S}$ will not necessarily have the address of $\mathscr{R}$, thus the $\mathscr{R}$ can advise; and the ontology agent of $\mathscr{R}$, $\mathscr{O}_{\mathscr{R}}$, has no duty to the $\mathscr{S}$ and so $\mathscr{R}$ must first request that it participate, which it could refuse to do. The *Negotiate Ontology* aspect divides into to two phases:

1. *Superconcept Phase* This makes use of the *FIPA-Iterated-ContractNet-IP* (see FIPA specification *SC30*,).
   (a) $\mathscr{S}$ makes a `cfp` (with action `inform-ref`, which requests that reference material, here a definition, be supplied) asking $\mathscr{O}_{\mathscr{R}}$ and its own ontology agent $\mathscr{O}_{\mathscr{S}}$ for a superconcept with the conditions that it satisfy a minimum set of attributes; in the form of *soft* constraints which are desirable and *hard* constraints which are required.
   (b) Step 1 may be iterated a number of times.
   (c) An agreement is reached or this stage fails.
2. *Admit or Assert Concept Phase* This makes use of *FIPA-ContractNet-IP* (see FIPA specification *SC29*).
   4. The $\mathscr{S}$ makes a `cfp` (with action `assert` $\langle$*concept*$\rangle$) to $\mathscr{O}_{\mathscr{R}}$ to admit `concept` into its ontology, `Ont`($\mathscr{R}$). The `cfp` is used to allow the receiving agent to respond with the proposal suggested or importantly to make a counter in the form of a different proposal.
   5. $\mathscr{O}_{\mathscr{R}}$ contacts $\mathscr{O}_{\mathscr{S}}$ to determine the proportion of concepts which have all the of the attributes of `concept` (called its *support*) in the ontology `Ont`($\mathscr{S}$), call this sup(`concept`).
   6. If sup(`concept`) $\geq T$, where T is some appropriate threshold, then $\mathscr{O}_{\mathscr{R}}$ takes steps to admit the concept permanently to `Ont`($\mathscr{R}$).
   7. If sup(`concept`) $< T$, where T is some appropriate threshold, then $\mathscr{O}_{\mathscr{R}}$ takes steps to make a temporary assertion of the concept using the constructs of *SL(n)* (see `www.fipa.org`), or other content language, as appropriate.

6 Discussion and Concluding Remarks

We have introduced the notion of an evolvable devolved ontology, a formal model which we have developed to address ontological structures and relations which arise in agile partnerships. We have shown how to use this to promote *ad hoc* semantic interoperability and thereby support communication. We believe that the parallels between agile partnerships and the context of the Pragmatic Web renders our approach, especially in the implementation discussed above, extremely useful to support the vision of the Pragmatic Web. Naturally, there are limitations and we briefly mention the most immediate. Using an FCA approach to (devolved) ontology structuring, as above, assumes that we agree upon a shared set of descriptors (e.g. attributes, properties) for objects of interest; which mean the same to each participant. Failure to achieve such an agreement would result in an empty common core ontology. There are ways in which to approach this, such as language games [SV97], semantic equivalence of terms, rough sets [Pa82]; however, we consider such discussion out of scope. We simply note that there are necessarily "levels" of agreement. A need for basic terms obtains in most approaches to ontology: cf. knowledge representation languages such as OWL (`www.w3.org`). Such descriptors tend to be fundamental and (often) "domain independent", e.g. colour, size, etc., and are usually quite easy to agree upon. More complex descriptors, if needed, can be derived from a preliminary application of our approach to align the semantics of these in terms of simpler descriptors. The point we wish to stress is that, contrary to most approaches, we require agreement only at the level of these basic descriptors. This reflects a pragmatic approach. Which descriptors pertain to descriptions of complex concepts remains a matter for negotiation among communicating parties. Importantly, we dispense with the need to agree upon structures for the domain of discourse, appealing instead to a "natural" structure. Structure remains in the actual application domains of the partners, but it is temporarily "forgotten" while a semantic alignment is sought[4]. Nor do we require any notion of "completeness" of definition of a complex concept in the core ontology. Such complete definitions can reside in the application ontologies with the concepts in the core ontology merely reflecting an overlap of "attribute-patterns". This provides for "approximation from below"[5], cf. [St02], allowing an agent to ground those concepts which emerge in the core from the "natural" structure, but which may not strictly exist as substantive concepts in its domain.
There are a number of related works to which we could make reference: this is to be expected in such an active area of research, rich with opportunities; for example, the applicability of FCA to ontology construction and management, is recognised by many, e.g., [St01] and [CHS04]. In the interests of space and brevity, we confine our-

[4] Furthermore, this is achieved rigorously and in a "reversible" manner through Galois connections [Bi67] and more generally through the categorical notion of adjunction [Ma98]. Explication of this would require the introduction of formal machinery which space prevents, thus we defer discussion to future presentations.

[5] For example, a car "approximates" a vehicle from below.

selves to three works of particular relevance. First, *Partially Shared Views* [LM90], which we have applied and discussed above. Second, *Exploiting Partially Shared Ontologies for Multi-Agent Communication* [St02], which exhibits some strong parallels in organising ontologies, for example, the author (independently) proposes an approach analogous to PSV. Third, the ontology merging techniques developed in *FCA-Merge* [SM01], offer a way in which to synthesise a global ontology (cf. Fig. 1) from application domain ontologies (cf. Fig. 2), in a manner analogous to *gluing* [GW99]. A recent state-of-the-art survey in the use of various formal mechanisms can be found in [KS03].

References

[Bi67] G. Birkhoff. *Lattice Theory (3rd Edition)*. AMS Colloquium Publication, Providence, RI, 1967.

[CHS04] P. Cimiano, A. Hotho, G. Stumme, J. Tane. Conceptual Knowledge Processing with Formal Concept Analysis and Ontologies. In *ICFA 2004*, pp. 189–207, 2004.

[DP02] B. A. Davey, H. A. Priestley. *An Introduction to Lattices and Order*. Cambridge University Press, second edition, 2002.

[Fe99] J. Ferber. *Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence.* Addison-Wesley, 1999.

[GFC04] A. Goméz-Pérez, M. Fernández-López, O. Cocho. *Ontological Engineering*. Advanced Information and Knowledge Processing. Springer, 2004.

[GG95] N. Guarino, P. Giaretta. Ontologies and knowledge bases: towards and terminological clarification. In N. Mars (ed.), *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*. IOS Press, Amsterdam, 1995.

[GW99] B. Ganter, R. Wille. *Formal Concept Analysis. Mathematical Foundations*. Springer-Verlag, 1999.

[KS03] Y. Kalfoglou, M. Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18: 1–31, 2003.

[Le83] G. Leech. *Principles of Pragmatics*. Longman, London, 1983.

[LM90] J. Lee, T. W. Malone. Partially Shared Views: A Scheme for Communicating Among Groups that Use Different Type Hierarchies. *ACM Transactions on Information Systems*, 8 (1), 1990.

[LR57] R. Duncan Luce, H. Raiffa. *Games and Decisions. Introduction and Critical Survey*. John Wiley & Sons, New York, 1957. Dover Reprint 1989.

[Ma98] S. MacLane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer, second edition, 1998.

[NK04] Natalya Noy, Michel Klein. Ontology Evolution: Not the Same as Schema Evolution. *Knowledge and Information Systems*, 6: 428–440, 2004.

[Pa82] Z. Pawlak. Rough Sets. *International Journal of Computer and Information Sciences*, 11: 341–356, 1982.

[Qu48] W. V. O. Quine. On What There Is. *Review of Metaphysics*, 2: 21–38, 1948.

[Sc60] T. C. Schelling. *The Strategy of Conflict*. Harvard University Press, Cambridge, Mass., 1960.

[SM01] G. Stumme, A. Maedche. FCA-MERGE: Bottom-Up Merging of Ontologies. In *IJCAI*, pp. 225–234, 2001.

[SMD06] M. Schoop, A. de Moor, J. L. G. Dietz. The Pragmatic Web: A Manifesto. *Communications of the ACM*, 49 (5): 75–76, May 2006.

[St01] G. Stumme. Using ontologies and formal concept analysis for organizing business knowledge, 2001.

[St02] H. Stuckenschmidt. Exploiting Partially Shared Ontologies for Multi Agent Communication, 2002.

[Su95] R. Sugden. A Theory of Focal Points. *The Economic Journal*, 105 (430): 533–550, May 1995.

[SV97] L. Steels, P. Vogt. Grounding adaptive language games in robotic agents. In I. Harvey, P. Husbands (eds.), *ECAL97*. MIT Press, 1997.

[SW86] D. Sperber, D. Wilson. *Relevance: Communication and Cognition*. Blackwell, Oxford, 1986.

[Wo02] M. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons Ltd, 2002.