Towards group decision making via semantic decision tables and blackboards

Cristian Vasquez and Yan Tang Demey

Semantics Technology and Applications Research Lab,
10G731, Vrije Universiteit Brussel, Brussels, Belgium
{cvasquez,yan.tang}@vub.ac.be

Abstract. Information infrastructures become increasingly decentralized. In these environments, coming up with coherent collective decision systems is difficult, especially in environments where central "authorities" are lacking. In this paper, we explore the idea of using artifacts called Blackboards for Decision Tables to support collaborative and incremental evolution of a network of Semantic Decision Tables, which we expect can improve the stakeholders capabilities to make decisions at a local and a community level.

Key words: Collaborative decision making, Emergent semantics, Semantic Decision Tables, Web Blackboards

1 Introduction

Within this paper, we aim to support a group of stakeholders in their decision making processes, in order to help them to accomplish their goals and maintaining an acceptable degree of coherence in their actions. In our scenario, the stakeholders may have their own inputs, processes and outcomes, which are modeled with business rules through explicit artifacts.

There are several decision support alternatives which aim to support these kinds of systems. Within this paper we will focus on the use of networks of simple but powerful artifacts called Semantic Decision Tables [6](SDT), which are conceived to deal with the ambiguity and conceptual reasoning difficulties that arise on large collaborative environments. Additionally, we make use of a representation mechanism called Web Blackboard to help incrementally convey shared understanding in decentralized environments.

Through this paper we will focus on how we can represent and interlink the local semantic rules of multiple stakeholders using artifacts which, we believe, can be useful for decision support in distributed environments.

We want to explore the possible benefits of an environment where multiple stakeholders (i) collaboratively define their rules and (ii) build up a network of semantic mappings, in order to reach configurations that support coherent decisions at community level.

This document is organized as follows: Section 2 presents a brief description of our problem. Section 3 will explore the related work on this subject. Section 4 will provide some definitions. Section 5 will describe how the decision artifacts are updated. Section 6 will describe the dynamics of this representation mechanism. Section 7 presents our conclusions and future work.

2 Problem Description

Suppose that we count with a group of citizens that move on daily basis to their workplaces. Some of them make use of public transportation consisting of three means: Bus, Tram and Metro. Thanks to shared schemas, this system counts with the necessary infrastructure to provide real-time information on public transportation, e.g. bus schedules, movements. Simultaneously, other citizens transport themselves using a car or bicycles, using information systems with similar characteristics.

Now let's suppose that due to the heavy traffic congestion, a third party provides a Peer to Peer (P2P) network where vehicle owners can offer theirs to others, specifying attributes such as vehicle model, vehicle consumption etc. In the same way, consumers can provide information such as schedules (i.e Use a car Mondays & Thursdays), collaboratively build promises of use etc. Within this system, users are allowed to define attributes on their own.

In this example, we show two scenarios with distinct semantic requirements. The former is a system that needs to integrate multiple information sources through the use of a global interchange schema, which can be built by a public transportation authority. The latter is a system that consists of multiple peers interconnected, where we don't necessarily have a central authority. In this scenario we may expect semantics that differ for each peer, for example users may have distinct representation of needs, cultural backgrounds, goals etc. To support the need of heterogeneous semantics in a P2P network, one possibility is to provide shared artifacts, where the peers incrementally try to convey structures to cover their Semantic interoperability requirements.

Naturally, this alone does not solve the collaborative decision making needs. We still need to provide an environment where the peers collectively build up configurations that support certain coherence at community level. Within the scope of this paper we will explore how the distinct peers can build up a network of SDTs in order to specify conditions to be processed globally. The goal of this is to come up with P2P decisions that are convenient to the community as a whole, for example to diminish the traffic congestion in a city.

3 Related Work

Because of their simplicity, Decision Tables (DT) are widely used tools to aid in decision making processes, providing reasoning in a compact form [3]. A DT is defined as a "tabular method of showing the relationship between series of conditions and the resultant actions to be executed", which like most programming languages associate conditions with actions to perform. Although DT decomposition and composition techniques allow us to scale into large DTs nets, they

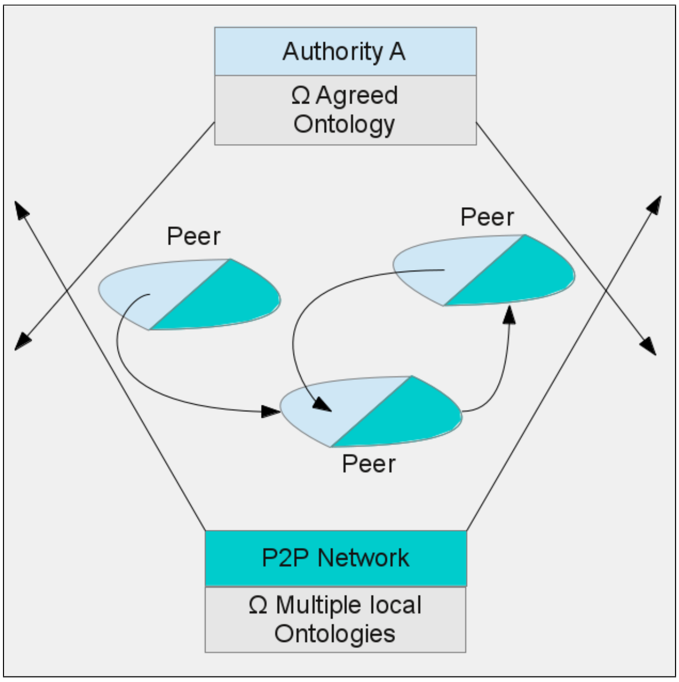

Fig. 1. A P2P network, with peers simultaneously interacting through global and local schemas. Global schemas are defined by an authority or group consensus, local schemas are incrementally constructed within a blackboard network

may become difficult to manage in environments where we count with heterogeneous decision makers, mainly due to misinterpretation issues. These ambiguity and conceptual reasoning difficulties can be diminished by the use of an extension of DT, called Semantic Decision Tables [7](SDT) that make use of shared and explicit decision semantics to maintain coherence between a net of distinct SDTs.

SDTs make use of ontologies, which are artifacts that facilitate information sharing and "understanding" between agents. In IT related domains, an ontology is understood as a shared, computer stored conceptualization in a formal language agreed upon a group of stakeholders that enables system interoperability [4].

How those ontologies can be constructed will depend heavily on the nature of the stakeholder ecosystem, and how difficult it is to reach global agreements. Depending on this, we can adopt distinct strategies to identify concepts (bottom-up, top-down or a combination). In this paper we will focus on systems where counting with "global interoperability" is difficult, and we regard interoperability as emerging from collections of incremental agreements between autonomous agents [2]. In this case, shared knowledge will be seen as the sum of all individual conceptions of the stakeholders. In order to support this we use artifacts called *Blackboards*, which can be seen as collective data spaces or playgrounds where the stakeholders can incrementally seek an acceptable degree of agreement about some topic of interest. The concept of blackboard is not new, they have a long research history [1]. They are artifacts conceived to facilitate the incremental construction of knowledge bases to be used by artificial intelligence applications. In the context of this paper we will use a variant called *Web Blackboards*[8], to support the stakeholder interaction with our SDTs.

Web Blackboards are dynamic data-spaces that can be traced along with their community of use. Stakeholders are free of creating and joining an arbitrary number of blackboards within their network, where they contribute to the models they describe in order to express for example, business rules. The architecture of such networks aims to provide a playground where local agreements are organically constructed in decentralized ways, building networks of blackboards via interlinking. The main characteristics of those networks are: (i) divergence and convergence capabilities in order to admit heterogeneity, and (ii) full traceability to support high-level interactions such as collective decisions.

4 System Design

In this section, we will provide some of the definitions that will support a system of this nature. We will provide definitions of some basic artifacts that will be used to collaboratively define a network to support collaborative decisions.

Definition 1 (Decision Table). *A* decision table *DT is a triple* (C, A, F) *where* (C) *is a set of conditions,* A *is a set of actions and* F *is a set of decision rules as defined below.*

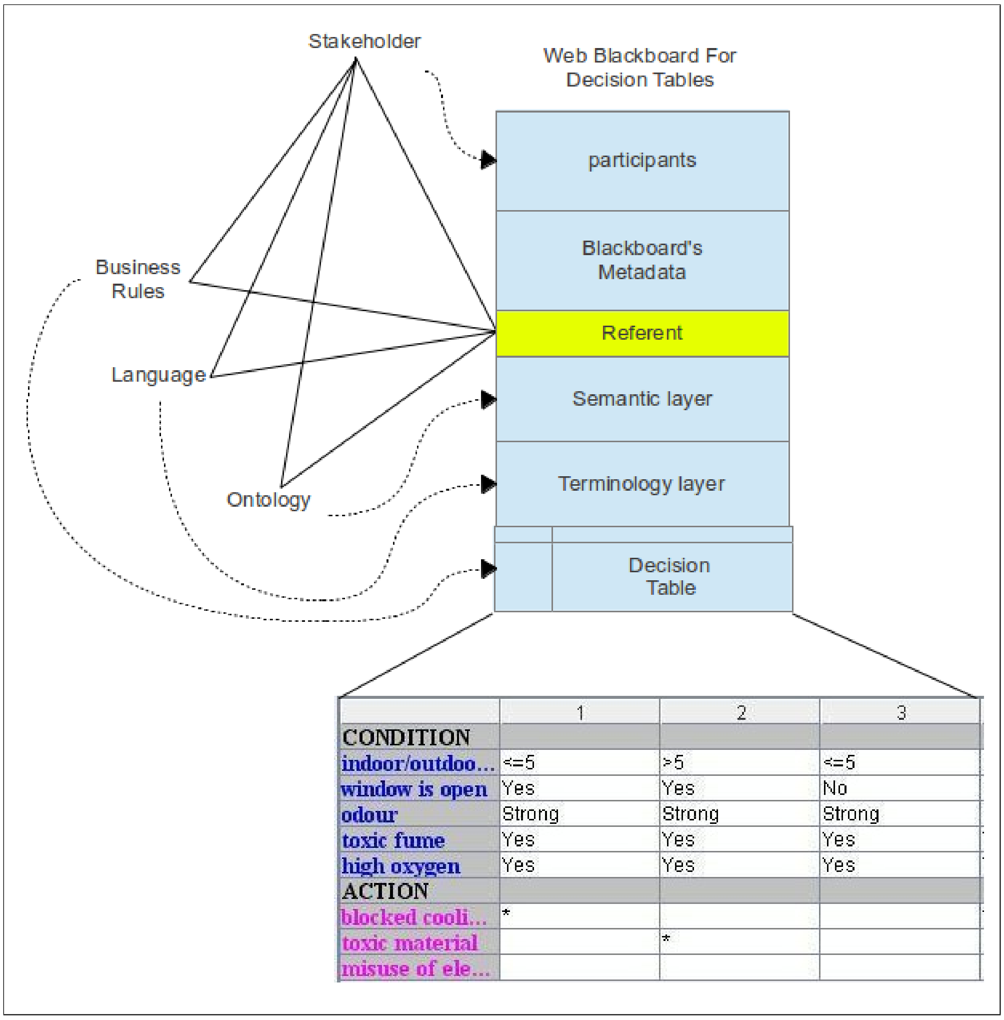

Fig. 2. Example of Blackboard for Semantic Decision Tables, where multiple stakeholders try to converge into a shared space, through distinct levels. Stakeholders incrementally modify terminology, shared schemas and rules.

– *A condition $c(c \in C)$ is tuple (cs, ce) where $cs(cs \in CS)$ is a condition stub (label) and $ce(ce \in CE)$ is a condition entry (a value or value range).*
– *An action $a(a \in A)$ is a tuple (as, ae) where $as(as \in AS)$ is an action stub (label) and $ae(ae \in AR)$ is an action entry (a value).*
– *A decision rule $f \in F$ is a function $f : CE^{cs} \rightarrow A$ where CE^{cs} as usual denotes the complete set of assignments of CS to CE.*

If we use L to denote the set of labels then we get $L = CS \cup AS$.

Definition 2 (Semantic decision table). *A semantic decision table SDT is a triple (DT, T, A) where DT is a decision table, T is a finite set of ontological annotation and A is a finite set of asserted axioms*

Given an ontology Ω and concepts $C_1, C_2, \cdots, C_k$ that are defined in Ω, an ontological annotation $t(t \in T)$ is a relation between a label $l(l \in L)$ and $Ci(1 \leq i \leq k)$, and denoted as $t(l, Ci)$

t is used (but not limited) to describe the following semantic relationships: instance-of/type-of, subtype/supertype, equivalent.

We can further specify A into two parts and denote it as $A \supset A' \cup A''$, where A' is part of ABox of domain ontologies that are used for the annotation and A'' is a set of axioms describing the meta-information of the DT. We get A' by searching the relevant axioms in the ontology using T.

Now let's look at a generic definition of blackboard

Definition 3 (Blackboard). *A model space ϵ is a finite set of models. Given a model space ϵ, a set of participants p, a shared ontology Ω and a uniform resource identifier URI, a blackboard B is a quadruple (e, p, Ω, URI).*

ϵ can be any imaginable and implementable model, such a data model, decision model, a rule or business model. In this paper, we will use semantic decision tables as the studied models. URI is a uniform resource identifier

A blackboard can be used for constructing a single SDT, establishing dependencies between two SDTs or aligning two ontologies.

We will give the relevant definitions in definitions 4 and 5

Definition 4 (Blackboard for decision tables). *A blackboard for semantic decision table B_{sdt} is defined as a quadruple (SDT, p, Ω, URI) where SDT is a semantic decision table that is represented on the blackboard, $p = p_1, p_2, \cdots, p_n$ is a set of participants, Ω is the shared ontology, URI is a uniform resource identifier*

We use a dot to indicate the owner of an element, e.g., if we have a semantic decision table SDT_1, then the condition stub set from SDT_1 is denoted as SDT_1. CS.

Definition 5 (Blackboard for specifying SDT dependencies). *Given two semantic decision tables SDT_1 and SDT_2, a blackboard for specifying SDT dependencies $B^{\widehat{sdt}}$ is defined as a quadruple (SDT, p, Ω, URI) where SDT is a set of dependencies between SDT_1 and SDT_2, p is a set of participants, Ω is the shared ontology and URI is a uniform resource identifier. Possible dependencies are illustrated as follows:*

- d_1*: label mapping, which is a mapping instance between* SDT_1.L *and* SDT_2.L*; A dependency of label mapping is denoted as* $d_1(d, d')$ *where* $d \in SDT_1$.L *and* $d' \in SDT_1$.L
- d_2*: input and output mapping, which is a* $1:1$ *mapping instance between* SDT_1.CS *and* SDT_2.AS*, and between* SDT_1.AS *and* SDT_2.CS*; A dependency of input and output mapping is denoted as* $d_2(dc, da')$ *and* $d2(da, dc')$ *where* $dc \in SDT_1$.CS*,* $da' \in SDT_2$.AS*,* $da \in SDT_1$.AS *and* $dc' \in SFT_2$.CS ...

5 Decision Tables change operators

How to manage changes within SDTs may depend largely on the dependencies between them. Take the following example, where SDTs are defined by different stakeholders (or group of stakeholders). As shown in Fig. 3, Stakeholder 1 commits to SDT_1 and SDT_2, Stakeholder 2 commits to SDT_1, SDT_2, SDT_4 and SDT_5, while Stakeholder 3 commits to SDT_5 and SDT_6. It is common that SDTs are dependent on each other no matter whether or not they come from the same organization.

When a stakeholder wants to make use of a SDT, he may join to the blackboard that holds its representation, if this stakeholder wants to make modifications there are two ways - one is to clone the SDTs and directly perform the modifications within the new variant, or start a modification agreement dialog with the other stakeholders that commit to the same SDT.

Each time that a B_{sdt} is cloned, we will have a new derived B_{sdt} that will be the result of a change with respect to the previous one via a set of *change operators* applied sequentially by the stakeholders. Below is a list of preliminary valid change operators within a single SDT variant.

α_1 : rename condition stubs, condition entries, action stubs and action entries
α_2 : update decision rules
α_3 : update condition/action entries
α_4 : remove conditions and actions
α_5 : add conditions, actions and decision rules
α_6 : add new axioms
α_7 : reinterpret the table content by annotating with another domain ontology
α_8 : update dependency relations

The choice of the change operators to be used in each group decision system will depend on the application requirements and is a matter of design. For example some applications may use very granular ones, such as `update decision rule` or `add new axiom`, while others may prefer execute change operators such as `extend bus schedule` which can be constructed via composition of more granular ones via layered operator frameworks [5]. The design of the change operators to be used within the B_{sdt} network will depend on usability and complexity trade-off decisions. Conceptual operators are important to maintain consistency in each ontology change and their underlying representations and to map them to verification and validation processes [4].

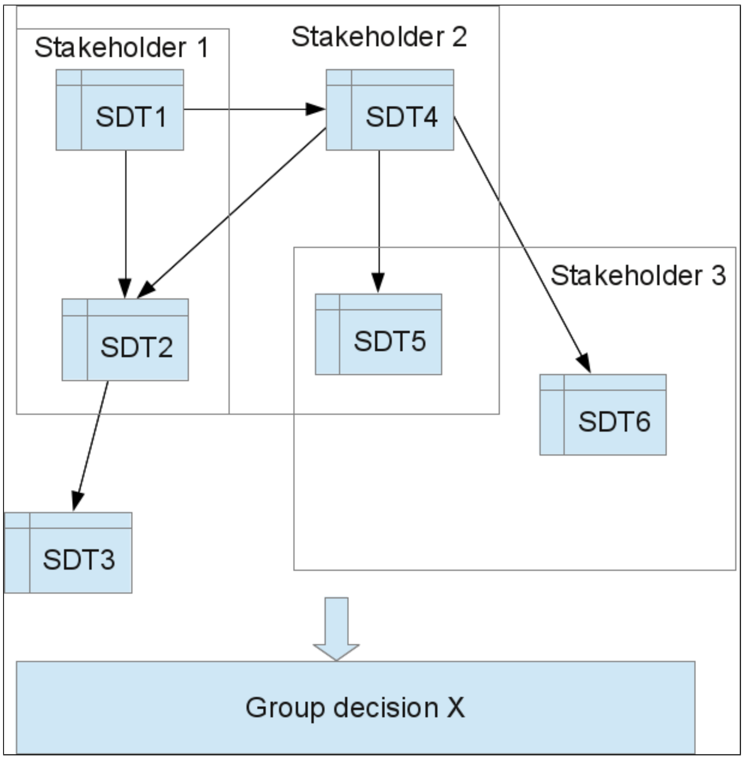

Fig. 3. Dependency graph of a group of SDTs. We use narrow arrows to indicate the dependency between two SDTs. For example, SDT_4 is dependent on the outputs of SDT_2, SDT_5 and SDT_6, and in the meanwhile, it provides outputs to SDT_1.

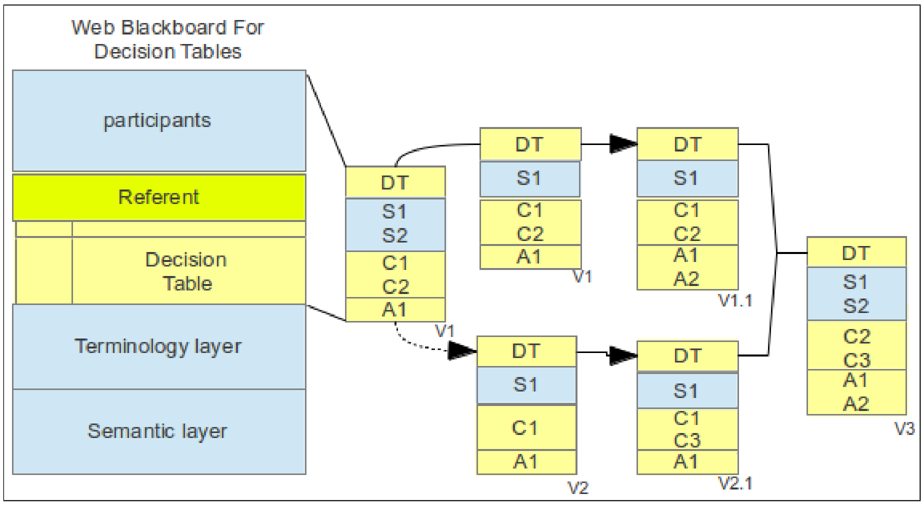

Fig. 4. Example of divergence and convergence of a Blackboard variant. Stakeholders, Stubs, Conditions and actions are traced in each mutation

6 Blackboard For Decision Tables Evolution

In the same way, the incremental and collaborative change in a network of B_{sdt} will result in a increasing map of variants that is a network.

In Fig. 4 we present a simplified diagram that shows how a small set of stakeholders deal with multiple B_{sdt} variants. In this situation, Each B_{sdt} variant is identified globally and can be branched from an ancestor B_{sdt} or merged with other branches, defining distinct sets of conditions and actions. Each time that a ancestor variant merges or branches, the applied sequence of change operators is registered and identified forming a traceability unit that may include pointers to deltas between two states, discussions between the involved stakeholders, updates to dependencies between Decision Tables etc. Some operations may not need branching or merging of B_{sdt} variants, such as terminology renaming, which can be considered as annotations of the underlying ontology.

To support these branching and merging capabilities, and to represent the variants evolution, we adopt a direct acyclic graph model (DAG). This DAG allows us to represent the i) branching an Bsdt variant ii) merging of two variants of with common ancestor. iii) recording the traceability information such as ancestors or sequence of change operators with respect to those ancestors.

This provides a strong support for non-linear SDT development with an approach that has already been proven successful in other fields (e.g., collaborative software development with version control systems such as GIT[1]. Within this approach we allow multiple SDTs to coexist, improving flexibility and scalability, but at the cost of a higher management complexity.

[1] http://gitscm.com

7 Discussion And Future Work

Through work, we explored the notion of decentralized SDT development through B_{sdt} networks. We expect that these kinds of layouts are useful to provide coherent decisions at community level in environments where we don't necessarily count with central authorities. This leaves us space for further questions, such as:

(i) What should be the agreement mechanisms between the stakeholders that commit to a B_{sdt}?
(ii) How we can profit from B_{sdt} networks to propagate, group and execute distinct SDT to improve collaborative decision making?
(iii) How these networks evolve, and how they can be observed by the stakeholders increasing their awareness?
(iv) How we can profit from the traceability of B_{sdt} networks, to provide insight that support cooperative social "order"?

7.1 Acknowledgments

The research described in this paper was partially sponsored by the INNOViris Open Semantic Cloud for Brussels (OSCB) project.

References

1. DD Corkill. Blackboard systems. *AI expert*, 6(September):40–47, 1991.
2. Philippe cudr E-mauroux. Emergent Semantics: Rethinking interoperability for large scale descentralized information systems. 2006.
3. Y Demey and TK Tran. Using SOIQ(D) to Formalize Semantics within a Semantic Decision Table. *Rules on the Web: Research and Applications*, (D):224–239, 2012.
4. Thomas R Gruber. A Translation Approach to Portable Ontology Specifications. *Academic Press Ltd. London, UK*, 5(April):199–220, 1993.
5. Muhammad Javed, Yalemisew M Abgaz, and Claus Pahl. A Layered Framework for Pattern-based Ontology Evolution. *Framework*, 2011.
6. Yan Tang. *Semantic Decision Tables - A New, Promising and Practical Way of Organizing Your Business Semantics with Existing Decision Making Tools*. LAP LAMBERT Academic Publishing AG & Co. KG, Saarbrücken, Germany, 2010.
7. Yan Tang and Ioana G. Ciuciu. Semantic Decision Support Models for Energy Efficiency in Smart-Metered Homes. *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 1777–1784, June 2012.
8. Cristian Vasquez. Blackboard Data Spaces for the Elicitation of Community-based Lightweight ontologies. In *IEEE/ACM ASONAM 2012*, 2012.