Rule-based Business Policies Representation, Reasoning and Integration in an Enterprise

Naeem Khalid Janjua 1, Farookh Khadeer Hussain 2

Digital Ecosystem and Business Intelligence Institute (DEBII)
Curtin University, Perth, WA

1naeem.janjua@gmail.com
2farookh.hussain@cbs.curtin.edu.au

Abstract—Business policies are pervasive in web-based information systems and the introduction of the semantic web has opened new fertile horizons for their representation and automated processing. However, a number of challenges have also emerged. One of those challenges is the contradictory information detection and resolution among policies. This papers reports on a semantic, knowledge-based approach for business policies representation, conflict detection and argumentation based resolution, leading to a justifiable conclusion. We also propose an architecture for enterprise-wide information system to integrate business policies about a given subject spanning across different information systems in an enterprise for decision making purposes.

I. INTRODUCTION

The WWW, being universal interface and underlying infrastructure for Intelligent Web Information System (IWIS) [26], has revolutionized the design, development, deployment and usage of information systems. Businesses now rely on the internet to conduct a wide range of activities, including buying and selling products, supporting web of relationships between a company and its employees, manager, partners, customer and researching and analysing development opportunities. As a result, the Enterprize-wide information systems have become a new trend in DSS research [20].

Today, the enterprise environments are becoming complex, competitive and dynamic and it demands for the organizations to be more flexible and responsive to environmental changes. Because of globalization, organizations are often compelled to form alliances with other organizations to survive and proposer. Therefore, it has become pivotal for enterprizes to understand their way of doing business. One important step in this understanding is represented by the identification and modeling of the business policies which could reflect changing business requirements, procedures or other constraints over a period of time to run their business activities. A business policy is defined as a high level directive that exists to control, guide and shape how an enterprize realizes it course of action used as mechanism to implement the business policies [16]. Business rules are derived from business policies and they provide the basic understanding how a business operates. For agile business environment managers always looks to take advantage of new opportunities and turn to business rules to implement change in organization working. Information technology has already learned the benefits of separating data by processing and managing data as an independent component of systems. A rules-extended development approach does exactly the same thing for business rules: by reducing the amount of code that needs to be written, it shortens the time necessary to implement change[12], [2], [10]. Therefore, for development of enterprise-wide information systems, it is important for business rule to be represented explicitly in declarative manner and automatically applicable.

The introduction of semantic web address the issues of the interoperability and self-describing semantics in information systems. The semantic web is seeking a universal medium for data exchange, i.e. classifying, packaging and semantically enriching information for support of data automation, integration, and reuse across various applications [22], [23]. The core of the semantic web, i.e. ontologies, meta-data and relations for performing inference with rules, is a source of seamless information integration of heterogeneous information sources.

To exploit the full potential of rule-based approaches, the business rules and the Semantic Web communities have started to develop solutions for reusing and integrating knowledge specified in different rule languages. Rule Interchange Format Working Group (RIF WG) and chartered it to develop an interchange format for rules in alignment with the existing standards in the Semantic Web architecture stack. The Rule Markup Language (RuleML), R2ML, SWRL are promising efforts in this regard [5]. Therefore, its evident that efforts are being made in development of Semantics web rules stack to provides enough features for representation of business rules on the Semantic web.

Although, the semantic web removed the hurdles of information integration within or across enterprise boundaries but still there are certain challenges needed to be addressed. As the enterprises are characterised by rapid development and change, so business rules will need frequent updates. Moreover, different experts (working in different departments of same corporation or in different corporations that are in collaboration in virtual space) may be involved in formulating business rules. Rapid changes to business rules by different experts will inevitably lead to conflicts among business rules. Additionally, company’s managers often make decision on minimal information currently available. Such decisions may be invalidated later when more information becomes available. This kind of behaviour is called as non-monotonic or defeasible. However, in the Semantic web, most of the studies have focused on use
of monotonic logics in layered development of semantic web which provides no mechanism for representing incomplete information and handling of conflicting information.

Therefore, the researches in information systems can take the advantage of research in AI for successful realization of enterprise-wide information system. In this article, we define a knowledge-based approach to definition of business rules in an enterprise. We extend defeasible logic programming (DeLP) for business rules representation and reasoning. We also proposed an architecture for business rules integration about a given subject in an enterprise.

The rest of paper is as follows: Section 2 literature review, section 3 case study, Section 4 knowledge-based approach to business policies representation and reasoning. Section 5 knowledge integration followed by conclusion in Section 6.

II. REVIEW OF LITERATURE

In an open computing environment, such as the World Wide Web or an enterprise intranet, various decision support systems are expected to work together to support information exchange, processing, and integration. However, DSS are usually built by different people, at different times, to fulfill different requirements and goals leading to Xue2009:

1) different supporting infrastructures
2) different syntactic representations of information
3) different schematic designs of information models
4) different semantics of information models.
5) conflicts among information, and the presence of incomplete information hinders its integration into information systems and afterwards knowledge integration at enterprise level.

Mostly, integration efforts have focused largely on the first four issues [27], [21], [17]. In this paper, we discuss the fifth issue which has received little attention in existing literature: semantic information and knowledge integration in the presence of incomplete and conflicting information.

Currently, the use of ontologies for semantic information integration can be viewed from two perspectives. Firstly, ontologies were introduced as a shared, explicit specification of a conceptualization of a domain. Therefore, ontologies lead to integration tasks to describe the semantics of information sources and to make the content explicit. This also focuses on the design and development of common ontologies that can be extended for more specific application domain specification. However, this will exacerbate the integration problem [18], [4], [6], [25]. Secondly, ontologies with extended rules are used for reasoning purposes. This involves an extension of ontologies with rules, where inference and reasoning are central to the process. Here, rules are defined on top of ontologies to infer new knowledge. The proposals for integration of rules languages and ontology languages can be classified by the degree of integration 159. Firstly, the hybrid approach is one where there is strict separation between the rule predicates and ontology predicates and reasoning is done by interfacing the existing rule reasoner with the ontology reasoner; whereas, with the homogeneous approach, both rules and ontologies are embedded in the same logical language $\mathcal{L}$ without making a prior distinction between the rule predicates and ontology predicates, and the reasoning single reasoner can be used for reasoning purposes.

Many DSS applications are built using the second approach, i.e. ontology with extended rules, to ensure the availability of integrated, high quality information for decision support. Broadly speaking, such approaches fall into two categories:

In the following sections, we discuss these categories in detail. In all these attempts, the systems integrate information through reasoning with the help of ontologies under certain assumptions including:

1) The given problem can be fully addressed with available information (solution to the problem lies within the available information). In order to elucidate it, let us consider an example. A department in an enterprise wants to improve its product and would like to make use of all the information it holds internally in order to adequately identify issues regarding product quality and improve the product’s quality. The department ignores any information outside its own boundaries.

2) The information or specification of business rules for decision-making is consistent. In other words, it is assumed that no contradictory rules will emerge during the decision-making process.

3) New information will be consistent with the already available information or specifications.

4) New information does not lead to retraction of previous conclusions.

In the existing literature, there is no research on enterprise-wide Web-DSS that addresses the aforementioned issues of information integration for intelligent decision-making.

A. Defeasible logic-based Semantic Web-DSS

This is the second category of Web-DSS having the capability of integrating information which could be incomplete and inconsistent. In this type of Web-DSS, the special types of rules known as defeasible rules are deployed to incorporate defeasible or non-monotonic behaviour in the system.

Dr Prolog [1] is a Prolog-based implementation for carrying out defeasible reasoning on the semantic web. It provides declarative system support rules, facts, ontologies, RuleML, and both monotonic and non-monotonic rules. The system provides a number of variants such as ambiguity blocking, ambiguity propagation and conflicting literals.

Dr-Device [14], [3]is CLISP-based defeasible reasoning implementation for information integration provided with a VDR-Device reasoning system. Compared to Prolog, Dr-Device supports only one variant for information integration, i.e. ambiguity blocking.

Sweetjess [11] is another defeasible reasoning system based on Jess and closely resembles courteous logic programs. It allows for procedural attachment and it implements only one reasoning variant. Moreover, it imposes a number of restrictions on the programs so that it can map on Jess.

Table I compares different defeasible logic-based semantic web implementations. In the context of semantic Web-DSS, these implementations have various limitations. Firstly, they
provide either data-driven or goal-driven reasoning. Data-
driven moves from current facts to a certain conclusion, whereas goal-driven reasoning is used to validate the con-
clusion with supporting facts and answer the user queries. However, in the case of semantic Web-DSS, both types of
reasoning are needed for information integration. The existing
proposed approaches cannot handle both types of reasoning for
information integration. Secondly, they define explicit (user-
defined) individual preferences among conflicting rules at compile time to resolve conflicts between them. The use
of these priorities is usually embedded in the derivation mech-
anism and conflicting rules are compared individually during
the derivation process. In such formalisms, the derivation no-
tion is bound to one single comparison criterion. However, the
semantic Web-DSS is a source of defeasible knowledge as it
is open by nature and subject to inconsistencies deriving from
multiple sources; therefore, it is not possible to define priorities
in advance among conflicting rules and even if priorities exist,
it is not appropriate to compare rules individually during the
derivation process. As a result, these systems provide limited
information integration and no knowledge integration at all.
Additionally, all of them provide to the end user only a
textual explanation about the integrated information, and the
integrated information results are not exportable in Argument
Interchange Format (AIF).

III. KNOWLEDGE-BASED APPROACH TO MODELLING AND
REASONING ABOUT BUSINESS POLICIES

In this section we describe a knowledge-based approach
to modeling business policies and reasoning about them in
order to resolve any conflicts if exists and produce a process
map. In our previous work [13][WI-paper] we have extended
the Defeasible logic programming (DeLP) with data driven
reasoning. Figure 1 depicts the proposed framework that is
knowledge-based approach to business policies modeling and
reasoning about them. The proposed framework performs
following important task:

1) Modeling of business policies
2) Argumentative reasoning
3) Production of process map
4) Export process map

In following sections, we discuss each of these task in detail.

A. Modelling of business policies

Usually enterprises either don’t capture business policies
formally or just document in natural language. Those docu-
dmented business policies are for only human consumption and
they can’t be directly translated to machine processable format.
This makes the procedure for determining which business
policy apply to a certain business process and thereafter its
execution is a very costly and cumbersome.

In order to support explicit, declarative specification and
execution of business polices we have devised a semantic tech-
nologies based approach that take policy documents as starting
point and with help of a domain expert, the domain knowledge
is modelled in form of an ontology. Ontologies are good for
application independent domain knowledge representation.

Figure 2 depicts the graphical representation of ontology
with classes, sub-classes and relationships of case study dis-
cussed in section #.

We define travel ontology comprises of four classes, namely:
1) Task : Represent some activity, e.g in case study Book-
ing is task which needs certain action.
2) Resource: The enterprise resource needs to accomplish
some task e.g. creditcard.
3) Dataitem : The information required for execution of the
task e.g RFT forms.
4) Constraint : Constraint over tasks, resource of data item.

We extend the the travel ontology by defining instances of
case study. The concepts used in ontology have data properties.
For example traveller is an instance of class Resource having
data property name. Similarly, submit is instance of Task class
having data property PersonName and FormName. Similary,
we define data property for reach individual in an ontology.

B. Business rules specification

The next step is business rules specification. Defeasible
logic programming (DeLP) is used to represent the and rea-

<table>
<thead>
<tr>
<th>Language</th>
<th>De-Prolog</th>
<th>De-Device</th>
<th>Situated Courteous logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic</td>
<td>De-reasonable logic</td>
<td>De-reasonable logic</td>
<td>Situated Courteous logic</td>
</tr>
<tr>
<td>Semantic data</td>
<td>RDF/SWOFL</td>
<td>RDF</td>
<td>DAML+OIL</td>
</tr>
<tr>
<td>RuleML</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Incomplete knowledge representation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Conflict representation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Data-driven reasoning</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Goal-driven reasoning</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Conflict resolution</td>
<td>User defined individual preferences</td>
<td>User defined individual preferences</td>
<td>User defined individual preferences</td>
</tr>
<tr>
<td>Explanation</td>
<td>Textual</td>
<td>Textual</td>
<td>Textual</td>
</tr>
<tr>
<td>AIF reification</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Information integration</td>
<td>Limited</td>
<td>Limited</td>
<td>Limited</td>
</tr>
<tr>
<td>Knowledge Integration</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

TABLE I
COMPARISON OF DEFEASIBLE LOGIC BASED SEMANTIC WEB-DSS

Fig. 1. Framework : Knowledge-based business policies modelling
soning about business rules with certain extensions [ref] in KBS system.

Linking business rules to ontology-based business rules involves aligning the concepts and the relationships involved with concepts and relationship of an existing domain ontology. This improves the shared understanding of business rules and thus their reusability will be enhanced as required in open environment like web and enterprises [9].

There are two kinds of rules supported by the system. Firstly, strict rules and secondly defeasible rules. The strict rules are the rules in classical sense: when a rule’s conditions are true, apply the rules and get its conclusion. These rules are used to represent inference mechanism from conditions to conclusion without any doubt. Most of the time these rules are construct from terms like should be, must be, must and their opposite terms. Defeasible rules or refutable rules that link the set of conditions to a conclusion with certain doubt, that could be refuted by contrary evidence. This type of rule is indicated by words like usually, presumably, or sufficient or we could intuitively feel that it is refutable. Defeasible logic programming use argumentation mechanism to handle defeasible rules.

The business rules are defined on top of ontology. We used hybrid approach for definition of business rules. The predicates such as travel, booking etc defined in ontology are used for construction of rules.

For example The traveller instance of Resource class with data property will be represented in first order predicate form such as traveller(Name). Figure 3 depicts the business rules developed on top of travel ontology.

C. Argumentative Reasoning and process map generation

The Core of the proposed conceptual framework is a DeLP interpreter with argumentative inference engine capable of performing hybrid chain reasoning, i.e. both forward chain and backward chain reasoning. DeLP interpreter is responsible for interpreting the DeLP program and passes it to the inference engine to perform hybrid reasoning [ref][ref]. Here we describe the reasoning process briefly.

Figure 4 depicts the comparison of inference mechanism in standard rete based kbs and extended kbs with argumentation semantics. The standard rete contains conflict resolution strategy in order to maintain the working memory consistant. whereas in extended rete there is no conflict resolution strategy at time of matching phase and all the rules whose conditions are true are executed and new inferred facts are added to working memory. Additionally to handle inconsistency, we used argumentation semantics. The executed rules are considered as arguments as they support some conclusion on basis of some facts or evidence. The Rete algorithm involves two steps. The first is the compilation of rules in the form of a network and then to identify all rules whose left hand-side pattern match the working memory elements. Then the data sieve through the compiled rules base, and the number of rules will become a member of an active set of rules and result in a conflict set.

The argumentative reasoning is performed to come up the process map. During this arguments have conflicting conclusion... counter-argument, static defeat, dynamic defeat etc.

Figure 5 depicts the process map which is outcome of reasoning over business rules. We used the same. The process map is a reasoning chain. We extend the representation of reasoning chain to represent process map. The Tasks are depicted rectangle shape, the resources are depicted with round shape, DateItem are depicted as diamond and constraint are depicted as double line. The defeasible inference is depicted with dotted lines and strict inference as a line.

Constraints are also modeled as counter-arguments.
Conflict are considered as counter-argument and dialectical tree are constructed to identify resolution.

The system support two variants: The execution process map: The entire process map: By default all arguments are considered true. All predicates are initialized with basis values.

IV. ARCHITECTURE FOR BUSINESS POLICIES INTEGRATION

A. Recommendation construction and publication

B. Argumentation Schemes and valuation of recommendations

During process of argumentation, relationships among the arguments link them with each other in a certain pattern to support the ultimate conclusion. Such linking patterns called as Argumentation Schemes. Argumentation schemes provide a way to perform reasoning over set of premises and conclusion. These argumentation schemes have emerged from informal logics. Schemes help categorize the way the arguments are build and aim to fill the gap between logic based application and human reasoning by providing schemes capturing stereotypical patterns of human reasoning e.g. argument from expert opinion scheme. Formally, an argumentation scheme is composed of a set of premises $A_i$, a conclusion denoted as $S$, and a set of critical questions $CQ_i$ aimed at defeating the derivation of the consequent [7], [8].

C. Conflict identification and resolution among recommendations

D. Integration of business policies

E. Export for integration purposes

Ontology to model the context / vocabularies

Ontology diagram will come here: Task, resource, constraint, Dataitem, TaskReRelatinship, TaskConstraintRelationship.

Unfortunately the Decidable fragment of first-order-language that can be expressed with OWL-DL is too limited to Rules to model the, there Horn rules like SWRL but not defeasible therefore DeLP based rules for capturing defeasible nature and argumentation formalism to resolve conflicts [7]

V. CONCLUSION

Business Rules offer a flexible, adaptive approach towards applications development on a high level of abstraction, declarative rules languages have already been developed by different communities and deployed variety of semantic web application as well as in traditional IT systems. Companies manage and specify their Business logic in the form of rules. Rules are also being used for modeling business policies in cooperative systems, and they are gaining popularity as a means of reasoning about Web data. To exploit the full potential of rule-based approaches, the business rules and the Semantic Web communities have started to develop solutions for reusing and integrating knowledge specified in different rule languages. Rule Interchange Format Working Group (RIF WG) and charted it to develop an interchange format for rules in alignment with the existing standards in the Semantic Web architecture stack. The Rule Markup Language (RuleML), R2ML, SWRL are promising efforts in this regard [5]. Therefore, it’s evident that efforts are being made in development of Semantics web rules stack to provides enough features for representation of business rules on the Semantic web. Another approach widely used to capture complex business semantics is ontology-based approach [40], which seeks to model basic business logic and meta-knowledge about business domain using ontologies. While the rule-based approach mostly focuses on the operational procedures of a business model, the ontology-based approach serves the purpose of capturing the rationale of the underlying business logic as well as providing means for business models interoperability.

Representation of security policies and business policies by one language [15]

Automated support to enterprise modeling has increasingly become a subject of interest for organizations seeking solutions for storage, distribution and analysis of knowledge about business processes [11]. One of the most common approaches for describing business and the information used by that business is the rule-based approach [17]. It consists in identifying and articulating the rules that define the structure and control the operation of an enterprise [37]. The main expectation from automated solutions implementing this approach is the ability to automatically determine consistency of business rules in a business model.

Rule bases for describing complex processes may explode in size and become intractable by humans, thus resulting in modeling mistakes, or in the choice to useless formal approaches.
REFERENCES


