***Cover page***

***Manuscript title:*** *A Policy-based B2C e-Contract Management Workflow Methodology Using Semantic Web Agents*


***(Corresponding) First Author:***

**Kalliopi Kravari**
*Department of Informatics*
*Aristotle University of Thessaloniki*
*GR-54124, Thessaloniki, Greece*
*Email: kkravari AT csd.auth.gr*
*Tel: 00302310998231*
*Fax: 00302310998433*

***Second Author:***

**Nick Bassiliades**
*Department of Informatics*
*Aristotle University of Thessaloniki*
*GR-54124, Thessaloniki, Greece*
*Email: nbassili AT csd.auth.gr*

***Third Author:***

**Guido Governatori**
*NICTA, Queensland Research Laboratory,*
*GPO Box 2434, Brisbane QLD 4001, Australia*
*Email: guido.governatori AT nicta.com.au*

# A Policy-based B2C e-Contract Management Workflow Methodology Using Semantic Web Agents

**ABSTRACT** Since e-Commerce has become a discipline, e-Contracts are acknowledged as the tools that will assure the safety and robustness of the transactions. A typical e-Contract is a binding agreement between parties that creates relations and obligations. It consists of clauses that address specific tasks of the overall procedure which can be represented as workflows. Similarly to e-Contracts, Intelligent Agents manage a private policy, a set of rules representing requirements, obligations and restrictions, additionally to personal data that meet their user's interests. In this context, this study aims at proposing a policy-based e-Contract workflow management methodology that can be used by semantic web agents, since agents benefit from Semantic Web technologies for data and policy exchanges, such as RDF and RuleML that maximize interoperability among parties. Furthermore, this study presents the integration of the above methodology into a multi-agent knowledge-based framework in order to deal with issues related to rules exchange where no common syntax is used, since this framework provides reasoning services that assist agents in interpreting the exchanged policies. Finally, a B2C e-Commerce scenario is presented that demonstrates the added value of the approach.

***Keywords:*** Semantic Web, Intelligent Agents, e-Contracts, Workflows, Policies, Defeasible Reasoning.

## 1. INTRODUCTION

Over the last years, *Electronic Commerce* (e-Commerce) (Laudon and Traver 2012) is becoming the norm and it is certain that this will continue mainly due to the fact that users around the world increasingly use advanced technology. Yet, the numerous parties engaged in electronic transactions must be assured that the transaction is safe and robust. Hence, the prevailing e-Commerce legal tool is the e-Contract, a binding agreement between parties that creates relations and legal obligations. In essence, traditional contract principles and remedies also apply to e-Contracts. A typical e-Contract consists of a number of contract clauses with each of them to address some specific aspect of the overall covenant between the involved parties.

### 1.1 E-Contracts and Workflow Systems

E-Contracts overcome the delays and drawbacks of the manual process (Krishna et al. 2005) but they tend to be complex. Analyzing the transaction process, namely the execution of an e-Contract, researchers found that it involves several tasks which can be represented as workflows (Chiu et al. 2004). Following this principle, almost every enterprise in the area of contract management has adopted solutions based on workflows (e.g. (Exari 2013; ASC 2013)). A workflow is concerned with the automation of procedures, where information is passed between participants according to a goal (Daskalopulu et al. 2001). Hence, the workflow of an e-Contract must be carefully specified and related to meet the contract requirements. But having a well-formed e-Contract workflow is not so easy. There are some considerable challenges that could lead to poor contract management resulting in significant loss of resources and savings.

A main issue is that the majority of the available contracts are unstructured and manually prepared (Indukuri and Krishna 2010), leading to poor possibilities in reusing information to create contract clauses and notifications. Additionally, e-Contract benefits to users are not clearly articulated and communicated to them since the e-Contract terms are not widely visible. In essence, the lack of structured information ends up a domino effect; clauses preparation, negotiation and execution lacks efficiency, individuals are unaware of critical events and e-Contracts are not renewed or renegotiated on time. Another challenge is related to negotiation, the most important part in the overall process (Dunne et al. 2005). Despite its significance, there is an absence of clear metrics and measurements of e-Contract value and compliance. Hence, the parties face difficulties in negotiating and evaluating the contracts they are involved in.

So far, there are plenty of approaches that deal partially with e-Contracts. Usually, approaches are system or language dependent, limited to specific use cases or focused on the monitoring of contract execution, omitting the aspects of the process that leads to an e-Contract agreement (Karlapalem and Krishna 2006; Turban and King 2011). Additionally, although trust establishment is one of the most important aspects in commerce, there is a lack of approaches that take into account trust issues. Actually, the whole e-Contract lifecycle, including preparation, negotiation and acceptance is still handled manually. In many cases humans have not only to write and agree upon an e-Contract but also to translate manually into some computer-readable internal representation (Šaučiūnas 2012).

### 1.2 E-Commerce and the Semantic Web

Hence, e-Commerce seems quite vulnerable to multiformity and subjectivity, mostly due to the lack of standards that would provide the unifying syntax and semantics of information used in e-Commerce transactions and thus e-Contracts. Semantic Web (SW) (Berners-Lee et al. 2001) has the potential to provide the needed formalisms and technologies (Governatori and Pham 2009). Actually, using semantic web knowledge representation (e.g. RDF) and rules (e.g. RuleML) could facilitate e-contracting processes. The declarative, modular, and interoperable nature of the knowledge representation could lead to a relatively high degree of automated reusability in e-Contracts. Semantic Web rules provide a high level of conceptual abstraction, enabling automatic communication between heterogeneous systems since rules are interoperable.

More specifically, an e-Contract has to be interpretable by both humans and machines. Thus, ambiguity has to be eliminated and semantic dependencies have to be explicitly defined. In this context, currently, projects like Good Relations (GoodRelations Project 2013) are developing e- Commerce vocabularies that are already used by enterprises like Google and Yahoo! while industry colossi like Walmart (Gattani 2013) and Viacom (Degel 2013) have semantic technology in their sites. Furthermore, the complexity that is inherent in an e-Contract negotiation and its result has to be mapped accurately into the contract (Jertila and Schoop 2005). From a logical perspective every clause of a contract can be understood as a rule where there are the conditions of applicability of the clause and the expected behavior. Hence, an e-Contract can be represented by a set of (non-monotonic) rules.

### 1.3 E-Contracts and Intelligent Agents

Yet, the Web content needs not only to be expressed properly but also to be processed effectively. Intelligent Agents (IAs) (Hendler 2001; Berners-Lee et al. 2001) which benefit from Semantic Web technologies, performing complex actions on behalf of their users, seem a promising solution. Intelligent agents are able to comprehend relevant information and better satisfy task requests since they can perform tasks unsupervised, reflecting the specific needs and preferences of their users. As they are gradually enriched with Semantic Web technologies their use is constantly increasing. Tasks, such as negotiation and brokering services, can already be carried out by intelligent agents (Koppensteiner et al. 2011). Hence, using agent technology could be a promising approach for a flexible and efficient end-to-end e-Contract management.

Another strong motivation behind using intelligent agents for contract management is the fact that each agent is able to manage a private policy, a set of rules representing requirements, obligations and restrictions, and personal data that meet its user's interests (Stuart and Norvig 2009; Wooldridge 2009). Similarly, each e-Contract has a set of rules (clauses) that specify among others how it will be executed and outlines restrictions on the parties involved (Indukuri and Krishna 2010). In addition to the obvious similarity, agents are also able to address the need for flexible modeling and enactment, since they are not limited to data exchange. They can decide for themselves what actions to perform in order to meet their goal, resembling human interaction (Grosof and Poon 2004; Huang et al. 2010). Yet, agents do not necessarily share the same decision-making processes, communication languages, logic or rule representation formalism but they have to communicate somehow.

### 1.4 Our proposal

Taking the above and our previous study (Kravari et al. 2010a) into account, we propose a policy-based workflow management methodology for e-Contracts using semantic web agents, conferring significant benefits in terms of interoperability, reusability, automation and efficiency. In our first study (Kravari et al. 2010a) we focused only on the negotiation process, proposing an initial methodology without proposing the current process and mainstream goals. Since then, we evaluated, improved and enriched the methodology. The current proposed end-to-end e-Contract management solution deals with all aspects of an e-Contract, from the trust establishment among the parties to the final agreement and approval, and it can be easily adopted in any multi-agent system or even in the web services field.

Additionally, our current proposed approach takes into account strategy and performance evaluation which enables participants to study their own and the rest parties' behavior in order to gradually improve their acting, gaining greater benefits from their trade. Moreover, in order to deal with agent heterogeneity, this article presents the integration of the revised and updated proposed methodology into a multi-agent knowledge-based framework, called EMERALD (Kravari et al. 2010b). This framework deals with this issue proposing the use of trusted, third party reasoning services that can be used in safely exchanging policies with heterogeneous rule formalisms. This way the exchanged policies will be correctly interpreted by negotiating parties.

Finally, this study presents for demonstration and evaluation purposes a B2C e-Commerce contract negotiation scenario. Our intention is to focus on B2C transactions in order to explore how our methodology adds value to this wide used form of commerce by enhancing the efficiency and effectiveness of the trade. More specific, this scenario is applied to the tourism field, since the increasing complexity of tourism business models and tasks require a new generation of e-Tourism infrastructures to support automation, integration and collaboration. Currently, agent technology is considered amenable to be applied in the field, yet there is still a lack of a unified solution for tourism services (Gärtner et al. 2008; Zhang 2009; Park et al. 2012).

The rest of the article is organized as follows. In Section 2, we present e-Contracts and their lifecycle analyzing each stage's challenges along with promising solutions. Section 3 presents intelligent agents and their ability to adopt these solutions dealing with e-Contract challenges in a policy-based context leading to an efficient workflow management methodology.

Section 4 presents the integration of this methodology into the multi-agent knowledge-based framework. In Section 5, a B2C e-Commerce case study is presented that demonstrates the added value of the approach. Section 6 discusses related work, and Section 7 concludes with final remarks and directions for future work.

## 2. E-Contracts

E-Contracts, as already mentioned, are agreements between two or more parties to create legal obligations between them, which are modelled, specified and executed by a software system (Krishna et al. 2005). They consist of at least two parties, here agents, and a number of clauses. Each party possesses a well-defined role, specified in the e-Contract. For example, in an on-line e-Commerce transaction the parties involved may be defined either as buyers or as providers. Typically, an e-Contract is described by the abstract specification:

$$EC = \{P, C\}$$

where:

- $P$ is the set of parties involved $P = \{P_1, P_2, ..., P_n\}$, $n \geq 2$, and
- $C$ is the set of clauses $C = \{C_1, C_2, ..., C_m\}$.

Although there are major differences between B2C (Business-To-Consumer) and B2B (Business-To-Business) e-Commerce since they have profound differences in the partners' motivation and behavior, our approach can be applied to both of them with minor differences in implementation (Laudon and Traver 2012; Turban and King 2011). B2B e-Commerce is usually more complex and requires a higher level of security, authoring, payment and after-sale support than B2C (Hutt and Speh 2012). Generally, B2B involves five to ten more integration issues in comparison to B2C. Hence, we studied both and designed a general methodology that could be applied to B2B with 2 or 3 slight implementation extensions, such as higher level of security, and to B2C without any further needs (Gummesson and Polese 2009). Yet, we focus on the B2C case in order to explore how this large marketplace can be improved in terms of efficiency. For this purpose, our proposed methodology adopts novel, usually B2B-related, features that were neglected so far, like varied pricing strategies and automated evaluation management.

### 2.1 E-Contract lifecycle

E-Contracts have three core principles (Krishna and Karlapalem 2008):

- *They can be comprised of subcontracts.* Complicated e-Contracts can be represented as nested hierarchical contracts. These subcontracts usually are not able to exist on their own and they have to be part of another contract.
- *They specify the list of activities* that should be performed by the involved parties.
- *They will be valid for a specific period*, which should be also defined in the e-Contract.

Yet, from group to group there are variations in how the lifecycle is defined (e.g. Zycus 2013; Infosys 2013, Mawji 2005, Miles et al. 2008). Studying these viewpoints along with our efforts to model e-Contracts in our simulation use cases (one of these cases is presented in section 5) we noticed that there are five stages that must always be present in some form. These stages are authoring, negotiation, preparation, execution – compliance and renewal. Hence, in this study, we analyze and adopt them. Fig. 1 presents the e-Contract workflow stages, adopted in this study, from a high level point of view in BPMN format (BPMN 2012). Notice that BPMN stands for Business Process Model and Notation which is a standard for business process modeling. In BPMN models consist of diagrams constructed from a small set of core elements, categorized into three major groups called flow objects, connecting objects and swimlanes. Flow objects are circles, rectangles or diamonds that indicate specific events and activities. Flow objects are linked with connecting objects, which are lines usually with arrows that indicate process direction. Swimlanes are lines running lengthwise within a rectangle called a pool. They organize diverse flow objects into categories having similar functionality.

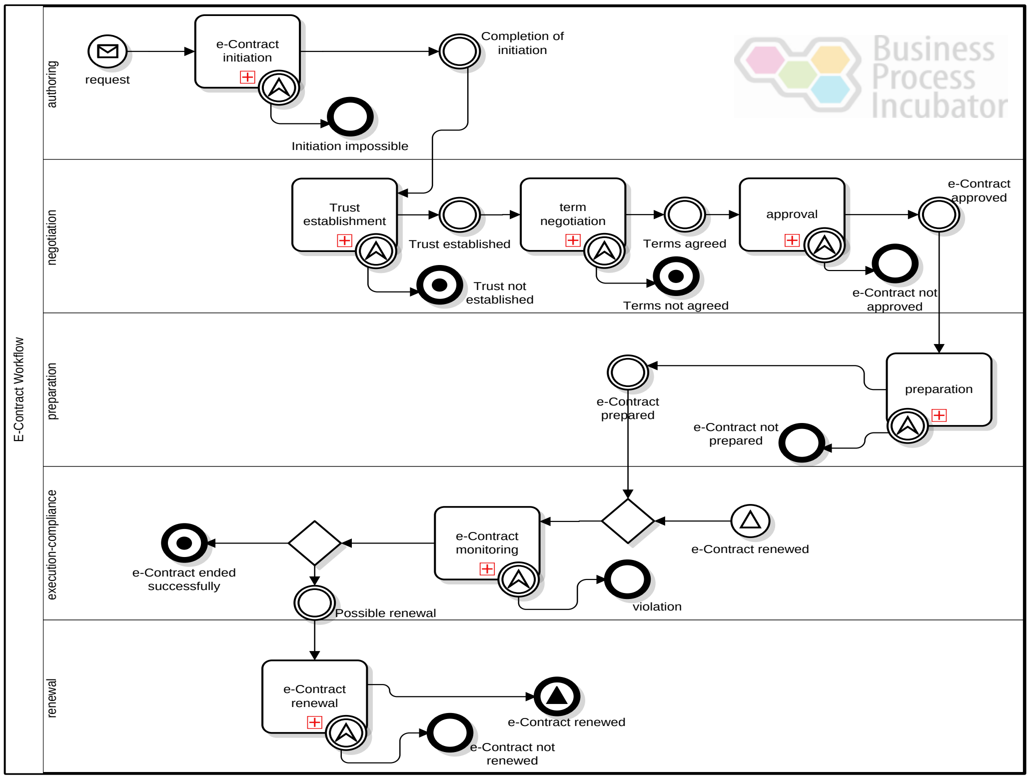


**Fig. 1.** E-Contract workflow stages.

In this context, Fig 1 describes the main procedures that are involved in each workflow stage. The first stage, called authoring, is related to e-Contract initiation. A successful initiation leads to the second stage, which includes trust establishment, term negotiation, if trust was established among the involved parties, and term approval in case of successful negotiation. After that, the e-Contract will be prepared (third stage). The next fourth stage is responsible for monitoring the execution of that approved and prepared e-Contract. As soon as, the e-Contact is completely executed, the parties will decide about renew it or not (fifth stage).

### 2.2 E-Contract lifecycle challenges

The first step in generating more accurate and profitable agreements is to provide an optimal authoring procedure. Unfortunately, this stage suffers from the traditional styles of management in which the majority of the contracts are unstructured and manually prepared. In order to deal with that, there is a variety of techniques and systems that extract data from manually prepared contracts (Indukuri and Krishna 2010; Stadermann et al. 2013). The point of this attempt is to provide rule-based knowledge representations, clause and event patterns in order to let parties leverage new e-Contracts by reusing this information. In this context, the so-called clause libraries or reference bases seem to be an appropriate solution for having the available data easily accessible (Patterson 2010) and thus they are adopted in this study.

Another important issue is the e-Contract negotiation, where parties communicate submitting feedback, comments and corrections. Hence, additionally to contract templates, contract interaction protocols are also vital for e-Contract negotiation (Krishna and Karlapalem 2008). Protocols choreograph the participants' interactions and therefore they have to be common resources (Wooldridge 2009). To this end, in accordance to the fact that all participants must comply with the same protocol in order to interact, a public library or repository containing the negotiation protocols is considered valuable and thus adopted in this study. Such a library or repository will ensure that protocols are well-defined and easily accessed by everyone.

Assuming that the involved parties interact properly using or not the above libraries, they may reach an agreement (final approval step). This is usually followed by the execution process ignoring further preparation issues. However, interpolating, among others, an extra preparing stage, as in this study, to an e-Contract's lifecycle is considered to lead to strategic advantages (Zycus 2013; Infosys 2013, Mawji 2005). In this context, better e-Contract finalization will be possible by using approved templates. Clauses will be optimized to allow streamlined legal review. Sub processes to capture digital signatures could be also added to this stage. Additionally, approved and signed e-Contracts could be electronically stored for future use, creating a historical reference base. At this point, payment and other independent systems could also be informed by proper notifications letting the parties move smoothly to the execution stage.

At the execution stage, full coordination of all activities execution, clauses satisfaction and e-Contract progress in general is needed. Additionally, full tracking of all contract related events combined with a notification process will really improve the execution management, letting parties always be aware of critical events. So far, the majority of the approaches dealing with e-Contract monitoring focus on terms violation (Krishna and Karlapalem 2008). Yet, over the last years, performance issues have been raised (Hensher and Stanley 2008); how valuable was the contract after all or how much it cost eventually are two of the many questions that need answer. However, the current limited integration of clear metrics and measurements of e-Contract value and compliance is still an open research issue. To this end, this study proposes such metrics in order to assist parties in improving their strategy and minimizing risks.

Finally, the last stage in the lifecycle refers to renewal issues. Sometimes the parties agree during the negotiation process for an automated extension (renewal) or an automated termination of the e-Contracts, sometimes they agree to renegotiate the e-Contract. In all cases this decision was made during the negotiation process. Of course, mechanisms for capturing unexpected events during the renewal process should be included in this final stage but since the decision itself was made during the negotiation phase it is considered a negotiation issue.

## 3. E-Contract Policy-Based Workflow Management Methodology

After studying the core challenges in an e-Contract lifecycle, we found that there are two critical points: the negotiation and the evaluation subprocesses. Taking into account that an e-Contract can actually be divided into stages of information exchange and negotiation which can be represented as workflows, as mentioned above, in this study we mainly focus on e-Contract negotiation and evaluation phases, adopting for the rest phases promising approaches found in the literature, proposing finally an end-to-end solution. More specific, we build our approach on the philosophy that an e-Contract can be divided into groups of tasks forming special categories; e.g. the stage in which the involved parties negotiate the terms of agreement by means of an offer and acceptance. Taking additionally into account that an e-Contract is actually defined by a set of rules (clauses) just like the strategy of an intelligent agent, we adopt the use of intelligent agents to deal with the end-to-end management of an e-Contract. The natural categorization of an e-Contract in stages combined with the autonomous and human-like behaviour of agents which can understand the e-Contract steps and organize their policy accordingly, seems promising. Finally, taking into account the above observations and the commonly used contract protocols, we will present in this article how these generic stages can be analyzed and combined with agent technology. Hence, firstly in this subsection, we briefly present agents and their utility in the setting. Next, we present our approach for an efficient negotiation and evaluation management and then our proposal for the e-Contract overall process.

### 3.1 Intelligent Agents

Intelligent agents could positively affect e-commerce in light of the exponential information load increase for buyers and providers. One of the benefits e-commerce can provide is lowered transaction costs. Yet, in order to achieve this goal much of the transaction processing procedures need to be automated. If closing the deal requires either negotiation, or search for information, or similar activities, it also requires intelligence, and thus provides a rich application area for agents. Next is the turnaround time. In some e-commerce applications, quick turnaround time is absolutely crucial. Again, agents are able to classify requests and quickly perform tasks, providing a high service level by significantly lowering the transaction volume.

Another issue is the importance of closing the deal. The ability to close a deal via an agent can be proven valuable for companies and customers, as agents can increase the efficiency of e-commerce transactions, they can also improve its effectiveness. Additionally, finding the lowest price is just as important as closing a deal. For instance, comparison shopping over the Internet has become one of the most popular applications for agent technology. Agents can make the search for the lowest cost almost effortless for the customer. (Turban and King 2011).

Generally speaking, agents are capable of actions based on information they perceive, their own experience, and their own decisions about which actions to perform. Agents involved in an e-Contract actually act on behalf of their users, thus, they have to contract an agreement efficiently without human intervention. In order to achieve this, each agent possesses arguments that describe its requirements, preferences and restrictions. These arguments usually include data and rules that comprise the agent’s policy and characterize its behavior. The policy of an agent determines, in particular, the way the agent selects its action based on the information it has, defining exactly what the agent can or cannot do. A careful consideration would reveal that these policies can, like e-Contracts, be divided into groups, such as personal data or restriction rules. Thus, taking advantage of this analogy could lead to an automation of e-Contract procedures.

Yet, the variety in representation and reasoning technologies is one of the main issues in agent interoperation. An intelligent agent does not necessarily have to oblige to other agents’ logic, nor is it essential for the agents to understand each other’s rule representation format. In fact, intercommunicating agents usually “understand” different (rule) languages. Thus, it will be essential not only to come up with an automation methodology for e-Contract procedures, but also to provide the suitable framework that will overcome the above issues.

### 3.2 Policy-based Workflow Management Methodology

A number of researchers, among them the authors of this article, were motivated by the understanding that e-Contracts have the potential to significantly improve B2C transactions. To this end, our previous study to this issue (Kravari et al. 2010a) revealed that the execution of an e-Contract indeed involves groups of tasks, which can be represented as workflows. Then, the study was focused only on the negotiation process, proposing an initial policy-based workflow methodology that tried to deal with this issue. Since then, we evaluated and improved the methodology, in order to overcome its drawbacks and provide an even more realistic approach as presented below. Next we studied the evaluation process, another core issue in e-Contract management, proposing a performance evaluation approach while we use this knowledge in the negotiation process. Finally, we studied how our approach can adopt really promising solutions in the rest lifecycle in order to provide a stable and reusable end-to-end management proposal using agent technology. The revised and extended methodology is presented in this article.

### 3.2.1 Negotiation Methodology

In our first attempt the process of an e-Contract negotiation agreement was divided in stages, which in turn were divided into steps (Kravari et al. 2010a). This primarily approach, as turned out in our following research, had to be updated and extended. By studying the whole e-Contract lifecycle we found that each of its stages can be divided in a number of sub stages. We came out with the finding that a negotiation process dealing with e-Contract agreement is actually a group of mainstream goals, each of them containing a number of tasks (sub processes) which in their turn contain a number of steps. Hence, we resulted in a hierarchical representation of what steps it takes to perform a task, which is prerequisite for a higher level goal. To this end, this article proposes a policy-based workflow negotiation methodology based on the above hierarchy and the fact that both e-Contract clauses and participants' policies can be divided following the hierarchy's principles. More specifically, we propose the specification of an e-Contract to be extended to an 10-tuple:

$$EC=\{P, C, N_G, G, N_{TSK}, TSK, N_{STP}, STP, C_{STP}, COND_{STP}\}, \text{ where:}$$

- $P$ is the set of parties involved $P = \{P_1, P_2, ..., P_n\}, n \geq 2$,
- $C$ is the set of clauses $C = \{C_1, C_2, ..., C_m\}$,
- $N_G$ is the number of goals,
- $G$ is the set of goals $G = \{g_s, 1 \leq s \leq N_G\}$(where $s$ is the $s^{th}$ goal),
- $N_{TSK}$ is the number of tasks for each goal $N_{TSK} = \{{}^sN_{TSK}, 1 \leq s \leq N_G\}$ (i.e. ${}^sN_{TSK}$ is the number of tasks for the $s^{th}$ goal),
- $TSK$ is the set of tasks for each goal $TSK = \{{}^stsk^l, 1 \leq l \leq N_{TSK}: g_s \in G\}$ (i.e. ${}^stsk^l$ is the set of $l$ tasks for the $s^{th}$ goal),
- $N_{STP}$ is the number of steps for each task $N_{STP} = \{{}^sN_{STP}{}^l, 1 \leq l \leq N_{TSK}, 1 \leq s \leq N_G\}$ (i.e ${}^sN_{STP}{}^l$ is the number of steps for the $l^{th}$ task of the $s^{th}$ goal),
- $STP$ is the set of steps for each task $STP = \{{}^sstp_k{}^l, 1 \leq k \leq {}^sN_{STP}{}^l: g_s \in G, {}^stsk^l \in TSK\}$ (i.e ${}^sstp_k{}^l$ is the set of $k$ steps for the $l^{th}$ task of the $s^{th}$ goal),
- $C_{STP}$ is the set of contract clauses for each step:

$$C_{STP} = \{{}^sC_k{}^l: {}^sstp_k{}^l \in STP \wedge {}^sC_k{}^l \subseteq C \wedge (\forall s,l,k,s',l',k', {}^sC_k{}^l \cap {}^{s'}C_k{}^{l'} = \emptyset \vee (s=s' \wedge k=k' \wedge l=l'))\},$$

- $COND_{STP}$ is the set of step transition conditions $COND_{STP} = \{{}^scnd_k{}^l: {}^sstp_k{}^l \in STP\}$, which decide if the transaction can proceed from one step to the next and it is part of the agent's internal policy.

The difference between contract clauses and transition conditions is that the former refer to conditions of the contract that are publicly known whereas the latter refer to conditions of the state of the workflow that are private to the contracting agents.

In general, goals represent the main issues (mainstream goals) of the overall e-Contract agreement process (e.g. trust establishment) whereas tasks refer to the negotiation subjects involved in the process (e.g. identity verification). Thus, each goal requires a group of tasks (negotiation subjects) that are under negotiation by the parties. Steps, on the other hand, represent individual actions (e.g. receive credentials). Thus, each specific task requires a group of steps in order to complete. Transitions between goals and tasks are sequential, but transitions between the task steps do not need to be sequential. This assumption is based on the fact that an e-Contract negotiation is considered successful if and only if each goal (issue) ended successfully. In other words, the parties involved have to proceed gradually from the first negotiation issue (first goal) to the last. Since tasks could even be considered as subcontracts, the above fact is also valid for them. Hence, both the e-Contract goals and their tasks should be processed sequentially. On the other hand, steps represent individual actions, which mean that they could refer to an optional or even repeatable action. For this reason, steps (in each task) are processed loosely sequentially, allowing any of them to be omitted or repeated. Finally, the parties involved in the negotiation process can disagree at any point, terminating the negotiation without agreement. Thus, in order to reach an agreement and eventually execute the e-Contract, each goal and consequently each task has to be successful. In other words, each mandatory step has to be successful, i.e. the set of clauses and conditions of each step should be satisfied.

Our methodology involves at each task (Fig. 2) three types of steps:

a) the exchange of the agent's $P_i$ clauses ${}_i^sC_k^l$ to the agent $P_j$,
b) the evaluation of the ${}_i^sC_k^l$ clauses using agent's $P_j$ personal data ${}_j^sD_k^l$, and
c) the exchange of the results/conclusions ${}_j^sE_k^l = \mathcal{I}({}_i^sC_k^l \cup {}_j^sD_k^l)$ of step (b) from agent $P_j$ back to agent $P_i$, in order to test $(\mathrm{I}({}_i^s cnd_k^l \cup {}_j^sE_k^l))$ if the clauses of the contract are satisfied so that the contract negotiation workflow can continue or not.

Notice that $i$ and $j$ stand for the $i^{th}$ and $j^{th}$ agent respectively while $s$ is the $s^{th}$ goal, $l$ is the $l^{th}$ task and $k$ is $k^{th}$ step while ${}_i^s cnd_k^l$ is the set of step transition conditions that decide if the exchanged results/conclusions $({}_j^sE_k^l)$ are satisfied in order to proceed to the next negotiation step.

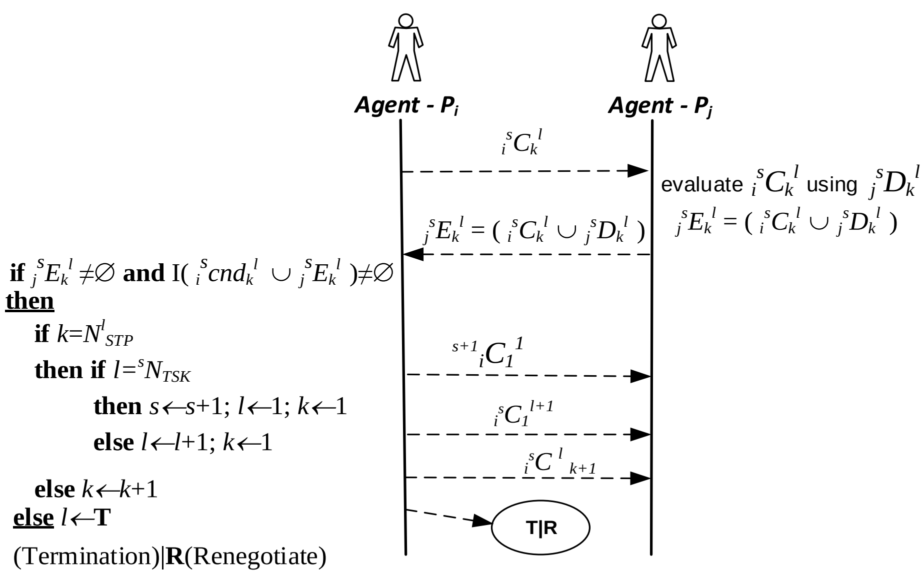


**Fig. 2.** E-Contract agreement workflow task steps

The workflow transition decision is taken by the following algorithm:

<u>*Transition Decision Algorithm (TDA)*</u>

**if** ${}_j^sE_k^l \neq \varnothing$ **and** $\mathcal{I}( {}_i^s cnd_k^l \cup {}_j^sE_k^l ) \neq \varnothing$

**then if** $k=N^l_{STP}$ */*if this is the last step of the task*/*

**then if** $l={}^sN_{TSK}$ */*and if this is the last task of the goal*/*

**then** $s \leftarrow s+1$; $l \leftarrow 1$; $k \leftarrow 1$ */*then move to the next goal*/*

**else** $l \leftarrow l+1$; $k \leftarrow 1$ */*else move to the next task*/*

**else** $k \leftarrow k+1$ */*else move to the next step*/*

**else** $l \leftarrow$ **T** (Termination)|**R**(Renegotiate)

In the case of irreconcilable disagreement the negotiation ends. In the case of solvable differences, the parties could renegotiate the running task, or the running goal, again depending on their personal policies. Whenever a goal ends successfully, it would not be renegotiated.

#### 3.2.1.1 Negotiation Mainstream Goals

Previous research has shown that each party involved in an e-Contract has to focus on a different set of contract terms during negotiation in order to maximize its profit. Hence, we studied the literature, comparing and combining the proposals of researchers and business experts in the field, such as (IACCM 2012; NCMA 2012), concluding to the most discussed and promising e-contract issues. The last five years thousand negotiation cases were studied revealing that what is negotiated is not what should be. While the economic impact of standard issues can be and it is negotiated, the biggest divide between trading partners today is in how they seek to allocate risk. Disagreements over acceptance or delivery are the number one cause of contractual claims and disputes. So it is not surprising that the parties focus so strongly on who will be liable for the consequences of failure. A quick review of the most negotiated issues reveals the extent to which negotiators are channeled into asset protection and the consequences of failure with intellectual property rights, governing law and payment terms being common examples. So far these issues along with price and payment overwhelmingly dominate negotiations. Table 1(a) presents the five most frequently negotiated issues during the last years.

**Table 1.** (a) most used negotiation issues - (b) proposed negotiation issues

| ***(a)*** *Most used negotiation issues 2008 - 2012* | ***(b)*** *Proposed negotiation issues* | |
|---|---|---|
| 1. Limitation of liability<br>2. Indemnification<br>3. Price, Charge<br>4. Intellectual property<br>5. Payment | 1. Change management<br>2. Responsibilities of the parties<br>3. Performance, Guarantees<br>4. Limitation of liability<br>5. Delivery, Acceptance | 6. Dispute Resolution<br>7. Price, Charge<br>8. Indemnification<br>9. Intellectual property<br>10. Payment |

Yet, effort in negotiations should be firstly placed in issues such as change management, responsibilities and guarantees. Parties should be asking why so many transactional activities fail to deliver to their potential and what role contracting and negotiation are playing in that failure. In this context, we propose a workflow for efficient e-Contract negotiation based on ten issues that are considered more productive in supporting successful relationships in a particular negotiation order (Table 1(b)). Although our proposal reflects what is considered efficient, the parties could omit or even add more goals. It is also worth mentioning that the most negotiation issues so far are still considered important but not as important as it is commonly believed in today transactions.

#### 3.2.1.2 Negotiation Workflow

To this end, we propose an eleven goal negotiation workflow. We propose eleven goals instead of ten (as in Table 1(b)), since we encounter an additional important issue, that of trust. Although the nature of trust in technological artifacts is still an under-investigation topic, the aim of it is to help parties trust each other. Establishing trust is still mainly pertained to a problem of authorization and access control (Lee et al. 2007). Hence, this first negotiation goal (trust establishment) should include among others identity verification tasks, certification validation tasks or recommendations checking tasks, depending on the parties' policy. In this context, we propose a policy-based approach, based on a set of policies and credentials (digital certificates). Usually, credentials are sufficient when the agent is convinced either of the other agent's identity or its membership in a sufficiently trusted group, such as the *Better Business Bureau* (BBBO 2012). Thus, the agents involved should firstly participate in an identity verification task in which they exchange the appropriate credentials that will enable them to trust each other.

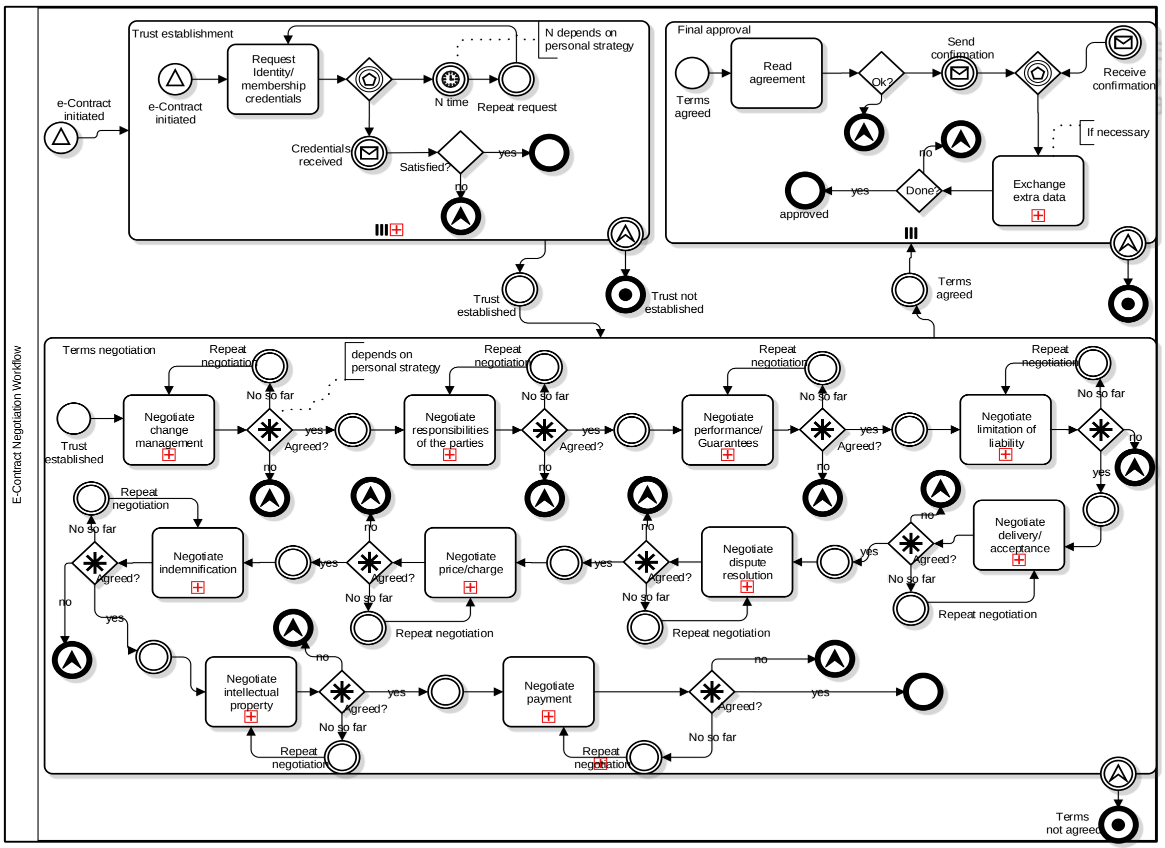


**Fig. 3.** An overview of the negotiation workflow.

An overview of the proposed workflow methodology is illustrated in Fig. 3. First, the parties have to deal with trust establishment, then they deal with the ten negotiation issues one by one and finally they conclude to final approval. There the e-Contract is approved and all the extra necessary data are sent. These data may include, among others, technical details, access and credentials, depending on the scope of the e-Contract. Hence, the parties are involved in tasks associated to reading agreement and closing the negotiation. It is worth mentioning that in our first attempt to provide a workflow methodology we provided a five stage workflow, including primary data exchange and contract monitoring (Kravari et al. 2010a). They actually should have been and are omitted since they are not parts of the negotiation process itself but they are processes that take place before and after the negotiation. Moreover, a terms of agreement stage was included that described the negotiation process in a high level whereas now that process is analyzed based on what issues should be negotiated in order parties to reach significant benefit.

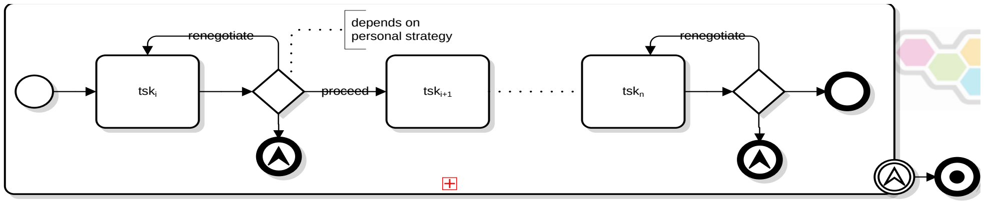


**Fig. 4.** An abstract view of the task workflow.

However, as already mentioned, during the negotiation any exception or irreconcilable disagreement might appear in any of the parties involved. In that case, the normal mode of execution in the workflow will be disturbed and thus the contract negotiation will be terminated. An abstract view of the task workflow is illustrated in Fig. 4 in order to help readers in understanding. The same principles can be adopted for the goal workflow too.

### 3.2.2 Evaluation Management

Evaluation is the systematic acquisition and assessment of information to provide useful feedback about some object. Most often, feedback is perceived as useful if it aids in decision-making. But the relationship between an evaluation and its impact is not a simple one. Studies that seem critical sometimes fail to influence short-term decisions, and studies that initially seem to have no influence can have a delayed impact when more congenial conditions arise. Despite this, there is broad consensus that the major goal of evaluation should be to influence decision-making or policy formulation through the provision of empirically-driven feedback.

In e-Contract management context, performance evaluation usually refers to e-Contract execution issues such as adherence to time schedule or quality of work. The accomplishment of a given task measured against preset known standards of accuracy, completeness and speed is indeed important. In a contract, performance is deemed to be the fulfillment of an obligation, in a manner that releases the performer from all liabilities under the contract. However, in this study we concluded that there is a need for evaluation not only for the performance but also for the negotiation strategies used by the negotiated parties. Hence, we acknowledge the need for two distinct evaluations, namely *strategy evaluation* and *performance evaluation*.

#### 3.2.2.1 Strategy Evaluation

Parties often wonder how they can evaluate their negotiation ability given the effect on their final benefit. Hence, in each negotiation not only boundaries must be set by parties but also goals must be measurable so achievement can be identified. In our proposal the term negotiation process is represented by ten goals plus one more for the trust establishment among the parties. These goals are measurable in the sense that measuring a goal has to do with how satisfied is each party with the result. Based on that and the fact that satisfaction is a purely subjective issue, we propose the use of a qualitative scale with five values, namely unacceptable, not satisfactory, insignificant, satisfactory and superior. Each party will be able to rate each result using one of these values whenever a negotiation process ends. To this end, we propose the use of a private *rating repository* where each party will store its ratings according with weight factors and estimations for each negotiation case, creating a database with its previous actions and their results.

Before defining the evaluation formula, it is important to clarify what should be rated. As already mentioned, negotiation is a procedure and as such it could be rated. Rating the final ascertainment is important, since each party expresses its overall feeling, but it is not enough. A bad decision could destroy the party's expectations but perhaps many other choices were right. Defining what was right and what was not is crucial in building a successful negotiation behavior. Hence, we propose a formula where not only the final result but also each negotiation goal is taken into account. Going one step further, in order to have a fully personalized evaluation, each party should be able to determine how important each goal is for him/her.

First we associate each qualitative value with a number (Table 2); values range from -2 (absolutely negative) to +2 (absolutely positive), with 0 indicating insignificance. We use both positive and negative values in order to reflect the satisfaction or dissatisfaction of a party as happens in real world negotiations. Next we insert weight factors in the formula (equation 1) in order to help parties define what is more or less important for them in a particular situation.

**Table 2.** Qualitative to Quantitative

| *Qualitative value* | unacceptable | not satisfactory | insignificant | satisfactory | superior |
|---|---|---|---|---|---|
| *Quantitative value* | -2 | -1 | 0 | 1 | 2 |

Equation 1 represents the final efficiency value (*EV*) for a negotiation based on ratings (see Table 2) for both the overall procedure (*Overall Rating – OR*, $OR \in [-2,2]$) and each individual goal (*Goal Rating – GR*, $GR_i \in [-2,2]$), e.g. payment negotiation. $CF_O$ and $CF_G$ stand for the user-defined category factors (*CF*). They are weights that define what is more important for the user, the final outcome ($CF_O$, $CF_O \in [0,10]$) or its components ($CF_G$, $CF_G \in [0,10]$), getting values from 0 (insignificant) to 10 (absolutely important). Additionally, each individual component (goal outcome) should also reflect its importance for the party, thus more weights (*Goal Factors – GF*, $CF_i \in [0,10]$) are defined.

$$EV = \frac{CF_O}{CF_O + CF_G} \times OR + \frac{CF_G}{CF_O + CF_G} \times \frac{\sum_{i=1}^{n} (GF_i \times GR_i)}{\sum_{i=1}^{n} GF_i} \qquad (1)$$

Hence, to negotiate strategically humans (and agents) need to learn to pick good strategies for negotiating with others, namely maximize their efficiency value (EV). Usually in a negotiation one side is winning and the other is losing, sometimes both sides win and rarely both sides lose. What determines who is winning, and thus satisfied, is the strategy. Yet, strategies vary from extremely competitive to extremely cooperative. In this context, how a party should pick the right one is crucial for its success. Moreover, a party is not always using a unique strategy during a negotiation, for instance he/she could use different negotiation strategies for different negotiation parts/goals. The question here is which is the optimal combination of negotiation strategies and where can we find them. To this end, following the previous philosophy (rating repository) we propose the use of a private *strategy repository* where each party will name and store its private strategies along with its rank.

We define as rank (R) the score of each used strategy, a newly stored strategy will get 0 points whereas each time a strategy is used the user's rating (GR) will be added to the rank value (R+=GR). It is worth mentioning that a strategy's rank may be reduced since GR could be negative ($GR \in [-2,2]$). Hence, using the information stored in both rating and strategy repositories a simple algorithm (presented below) could deal with the strategy decision issue. A variety of such algorithms could be used by an agent in order to build its negotiation behavior over time, depending on its user's personal perception. The proposed algorithm is simple but guarantees that it will always adopt the most promising strategy for each negotiation goal.

*Strategy Decision Algorithm (SDA)*

Define new (e-Contract) goals.
Define threshold x. */*refers to how important should be an already negotiated goal (GF)*/*
Find similar-case with max EV. */*max EV represents the best performed negotiation so far*/*
∀ goal ∈ similar-case
    If goal == new goal */*if similar goal was negotiated its strategy should be checked*/*
    Then get goal(GR) and goal(GF)
    If goal(GR)=2 and goalGF>x */*if the outcome was superior*/*
        Then adopt goal(strategy) for new goal */*its strategy is promising*/*
Else check rest similar-cases. */*having no superior outcomes the rest cases should be checked*/*
∀ similar-case
    ∀ similar-case(goal)
    If similar-case(goal) == new goal */*check each case for similar negotiated goals*/*
    Then get similar-goal(GR) and similar-goal(GF)
    If similar-goal(GR)=2 and similar-goalGF>x */*if found one with superior outcome*/*
    Then adopt similar-goal(strategy) for new goal */*adopt its strategy*/*
Else adopt strategy with max rank (R). */*if there is no superior outcome so far adopt the strategy with the maximum rank, it proved to be the best so far, thus it could be promising*/*

#### 3.2.2.2 Performance Evaluation

As already mentioned, performance evaluation refers to e-Contract execution issues. Hence, specific criteria based on preset known standards of accuracy, completeness, cost, and speed should be defined to ensure all work is completed within the requirements of the contract. To this end, throughout the duration of contract execution each party must monitor and control (monitoring-compliance stage) the other party's performance, that is, whether he/she is executing his/her works based on the terms of the agreed contract or not. More specific, each party must observe the agreed baseline schedule or any approved revisions thereof, use the proper resources which were agreed in the e-Contract, refrain from making and claiming liquidated damages for expenses that are not provisioned in the e-Contract, observe the agreed quality and perform the proper quality controls of his/her deliverables that are specified in the e-Contract, apply the procedures provisioned in the e-Contract to settle issues arising during its execution, see to the observance of the agreed change management and the enforcement of the applicable procedures in each case, and enforce the procedures provisioned in the e-Contract to identify, assess and handle the risks that may appear during the execution of the contract.

Once all contractual obligations have been completed a more objective judgment about the e-Contract can be formed. Thus, in this study we focus on a final overall evaluation. It is worth mentioning that using performance evaluation during execution is subject of extensive monitoring issues research, which is out of the scope of this study, the evaluation criteria though are quite the same. Our intention is to provide a useful but simple approach to judge an e-Contract's outcome; e.g. was it satisfactory or not, would it be beneficial to commerce with that party again, what should be negotiated more carefully next time. Hence, defining the criteria that will reflect a party's compliance with the terms and requirements of an e-Contract is essential. Two seem to be the core criteria categories, time and quality of work. Time is the rating of the timeliness of completion considering the actual completion date compared with the original contract completion date. Quality of the work is the delivered quality. At final completion the quality of the materials, equipment or services incorporated in the work must meet the requirements set out in the e-Contract's specifications. Each of these categories contains a number of specific criteria; adherence to plan, action on anticipated delays and plan maintenance for time while work appearance, thoroughness and accuracy of work and liaison effectiveness. Following the same principles as in strategy evaluation, we associate each qualitative value with a number (Table 2). Equation 2 represents the final performance value (*PV*), where $w_i$ is the user weight for each evaluation criterion ($cr_i$).

$$PV = \sum_{i=1}^{n} (w_i \times cr_i) / \sum_{i=1}^{n} w_i \qquad (2)$$

### 3.2.3 Overall process

The most important parts of the overall process are already discussed. In this subsection, we just summarize them in order to provide an aggregated representation. Using semantic web intelligent agents is a core decision that allows automation and thus end-to-end support for an e-Contract management. Fig. 1 presented the five hierarchical stages of an e-Contract; authoring, negotiation, preparation, execution – compliance and renewal. Starting from authoring, we propose the use of a clause library described previously in this article. This public library will contain existing clauses and old e-Contracts, agents will be able to use them in order to author efficiently a new e-Contract. Next, the involved agents should use a protocol library. Using this library, agents will be able not only to choose the most appropriate protocol for the transaction but also to act properly throughout the overall procedure. Additionally, agents should use their private repository in order to choose an appropriate strategy that could maximize their profit.

Next, agents will be able to proceed to the next stage; the negotiation stage which was extensively discussed above. After a final approval, a preparation should take place. In that stage, the agreed e-Contract or part of it is going to be stored in the clause library for future use. Additionally, a payment system (e.g. paypal) should be linked with the involved agents in order to be used subsequently. Finally, the strategy evaluation holds here, whereas performance evaluation is included at the end of the execution stage. Finally, renewal should or should not take place depending on the previous agreement. In case of a renewal possibility, the involved agents first check the results of the performance evaluation in order to decide whether it is worthwhile for them to proceed. Here we proposed a sample overall procedure, based on some core principles, which of course could be enriched based on each e-Contract case.

## 4. A KNOWLEDGE-BASED INTEGRATION FOR THE METHODOLOGY

In this section, we present the integration of our proposed e-Contract workflow methodology into an existed multi-agent knowledge-based framework, called EMERALD (Kravari et al., 2010b; Kravari et al., 2010c). EMERALD is based on Semantic Web and FIPA standards (FIPA Specifications 2002) and, thus, it maximizes reusability and interoperability of behavior between agents. This integration allows us to use and evaluate our methodology, making it practically available, while EMERALD is enriched. This enriched framework is suitable for modeling and managing e-Contracts and it can be used for various types of on-line transactions among agents.

### 4.1 The EMERALD Framework

EMERALD (Fig. 5) is built on JADE (Bellifemine et al. 2003). It supported so far the implementation of various applications, like brokering (Antoniou et al. 2007; Kravari et al. 2009a) and agent negotiations (Governatori et al. 2001; Kravari et al. 2009b; Huang et al. 2011). This framework provides, among others, a generic, reusable agent prototype for knowledgecustomizable agents (*KC-Agents*), consisted of an agent model (*KC Model*), a directory service (*Advanced Yellow Pages Service*) and several external Java methods (*Basic Java Library*). Agents that comply with this prototype are equipped with a Jess rule engine (JESS 2008) and a knowledge base that contains environment knowledge (in the form of facts), behavior patterns and strategies (in the form of Jess production rules).

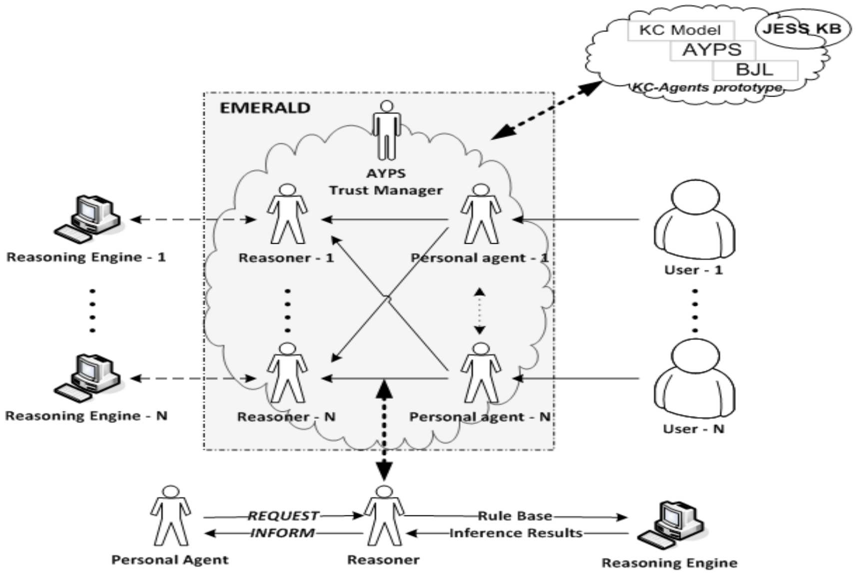


**Fig. 5.** EMERALD abstract architecture.

In this study, the Jess knowledge base represents the agent's *internal policy* and implements the workflow transition conditions ${}^{s}cnd_{k}^{l}$ of each negotiation step $k$ to the next $k+1$(for the $l$ task of the $s^{th}$ stage) (Fig. 2). This knowledge base is actually the agent's data and rules that comprise its policy and characterize its behavior, as described above. Using the KC-Agents prototype offers certain advantages, such as modularity, reusability, maintainability and interoperability of behavior between agents, as opposed to having behavior hard-wired into the agent's code. A short description is presented here for better comprehension.

First of all, the generic rule format for describing the agent's behavior is:

$$result \leftarrow rule\ (preconditions).$$

The agent's internal knowledge is a set of facts $F = F^u \cup F^e$, where:

- $F^u = \{fu^1, fu^2, \ldots, fu^k\}$ are user-defined facts, and
- $F^e = \{fe^1, fe^2, \ldots, fe^m\}$ are environment-asserted facts.

The agent's behavior is represented as a set of potential actions rules $P = A \cup S$, where:

- $A = \{a \mid fe \leftarrow a(fu^1, fu^2, \ldots, fu^n) \wedge \{fu^1, fu^2, \ldots, fu^n\} \subseteq F^u \wedge f^e \in F^e\}$ are the rules that derive new facts by inserting them into the KB, and
- $S = C \cup J$ are the rules that lead to the execution of a special action, such as:
  - agent communication $C = \{c \mid ACLMessage \leftarrow c(f^1, f^2, \ldots, f^p) \wedge \{f^1, f^2, \ldots, f^p\} \subseteq F\}$, or
  - Java calls $J = \{j \mid JavaMethod \leftarrow j(f^1, f^2, \ldots, f^q) \wedge \{f^1, f^2, \ldots, f^q\} \subseteq F\}$.

Moreover, the advanced yellow pages service (*AYPS*) provided by the *KC-Agents prototype* is a fully automated service for both registered services (services provided and advertised by an agent) and required services (specific services required by an agent). *AYPS* uses a repository to store both services, providing retrieval service. Concerning the *KC Model*, *AYPS* returns the providers as Jess facts with a designated format: *(service_type (provider provider_name))*. Additionally, as trust has been recognized as a key issue in Semantic Web multi-agent systems, EMERALD supports a variety of reputation mechanisms, both decentralized and centralized (Kravari et al. 2010b, Kravari et al. 2010d).

Another important issue in this framework is reasoning interoperability. Agents do not necessarily share a common rule or logic formalism, thus, it is vital for them to find a way to exchange their position arguments seamlessly. To this end, EMERALD proposes the use of Reasoners (Kravari et al. 2010c), which are actually agents that offer reasoning services to the rest of the agent community. This approach does not rely on translation between rule formalisms, but on exchanging the results of the reasoning process of the rule base over the input data. The receiving agent uses an external reasoning service to grasp the semantics of the rule base, namely the set of entailments of the knowledge base (Fig. 5). The procedure is straightforward: each Reasoner stands by for new requests and when it receives a valid request, it launches the associated reasoning engine and returns the results upon reasoning completion. Thus, although Reasoners are built as agents, actually they act more like Web services.

Currently, EMERALD implements a number of Reasoners that offer reasoning services in two reasoning paradigms: deductive rules and defeasible logic. Deductive reasoning is based on classical logic arguments, where conclusions are proved to be valid, when the premises of the argument (i.e. rule conditions) are true. Defeasible reasoning (Nute 1987), on the other hand, constitutes a nonmonotonic rule-based approach for reasoning with incomplete and inconsistent information, which is useful in many applications, such as business rules, e-contracting, brokering and agent negotiations. In the next subsection we briefly present defeasible logic background so that our use case (based on defeasible logic) to be comprehensible.

### 4.2 Defeasible logic

Defeasible logic has the notion of rules that can be defeated and it is part of defeasible reasoning (Nute 1987; Pollock 1992). Defeasible reasoning in contrast with traditional deductive logic, allows the addition of further propositions to make an existing belief false, making it nonmonotonic (Koons 2009). When compared to more mainstream nonmonotonic reasoning approaches, the main advantages of defeasible reasoning are enhanced representational capabilities and low (linear) computational complexity (Maher 2001; Antoniou et al. 2009).

One of the main interests in DL is in the area of agents (Governatori and Rotolo 2008). This is because, DL is a nonmonotonic logic and, thus, capable of modeling the way intelligent agents (like humans) draw reasonable conclusions from inconclusive information. This feature, which leads to more realistic conclusions and assessments, motivated plenty of researchers. Knowledge in DL is represented in the form of facts and rules. Facts are indisputable statements, represented either in form of states of affairs (literal and modal literal) or actions that have been performed. Rules describe the relationship between a set of literals (premises) and a literal (conclusion). Three types of rules are available strict rules, defeasible rules and defeaters.

- Strict rules take the form A1, … , An → B. They are rules in the classical sense: whenever the premises are indisputable then so is the conclusion. Thus, they can be used for definitional clauses.
- Defeasible rules take the form A1, … , An => B. They are rules that can be defeated by contrary evidence.

- Defeaters, finally, are a special kind of rules. They are used to prevent conclusions, not to support them and take the form A ~> p.

The main concept in DL is that this logic does not support contradictory conclusions, instead seeks to resolve conflicts. Hence, in cases where there is some support for concluding A but there is also support for concluding ¬A, namely where there are conflicting (mutually exclusive) literals, no conclusion can be derived unless one of them has priority over the other. This priority is expressed through a superiority relation among rules which defines priorities among them; namely where one rule may override the conclusion of another rule. For example, suppose that there is a conflict set {(c(x),c(d) | x≠d}; it consists of two competing rules r1 and r2:

r1: a(X) => c(X)

r2: b(X) => c(X)

where r2 could be superior (r2 > r1).

### 4.3 Integrating the Workflow Methodology into EMERALD

Following EMERALD's specifications we commit to Semantic Web and FIPA standards, thus, we use RuleML language (Boley et al. 2010) for representing and exchanging agent policies and e-contract clauses ${}_iC_k^l$. RuleML has become a de facto standard and it is very close to the RIF (Boley and Kifer 2010) standard for Semantic Web rules. In addition, we use the RDF model (RDF Specifications 2004) for data representation both for the private data ${}_jD_k^l$ included in agent's $P_j$ internal knowledge and the results ${}_jE_k^l$ generated during the negotiation steps. The overview of the above proposal is illustrated in Fig. 6.

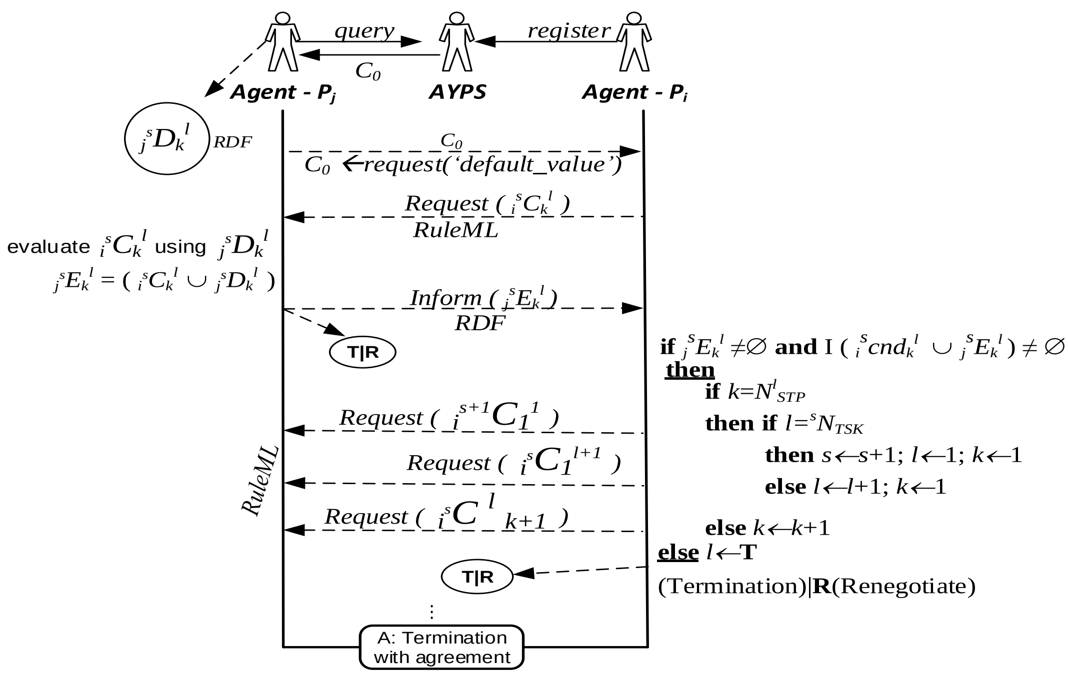


**Fig. 6.** Implementation of the workflow negotiation steps.

First of all, each (seller, provider, etc) agent should register to AYPS, providing among others the services or products it offers, the default call-for-negotiation value it requires, as well as its required formalisms/languages. Whenever an agent $P_j$ wants to find an appropriate provider for its need asks AYPS. AYPS replies with the appropriate list providing all the available information. Hence, the agent $P_j$, in order to start an e-Contract negotiation process with another agent $P_i$ (suggested by AYPS), sends a call-for-negotiation message (ACL message with *REQUEST* communication-act) to $P_i$ containing $C_0$, which is $P_i$'s default call-for-negotiation value; namely the appropriate information that a potential partner should send to $P_i$ in order to trigger its interest and receive back a positive response for negotiation.

Thereafter, the procedure proceeds as it was described in the second section. $P_i$ examines the new request and sends back a *REQUEST* message containing part of its clauses ${}_iC_k^l$ (in RuleML format), waiting for $P_j$'s reply or a termination (or a renegotiation request). $P_j$, on its behalf, evaluates the receiving ${}_iC_k^l$ clauses using its own private data ${}_jD_k^l$ (in RDF format) and informs $P_i$ with a new message (ACL message with *INFORM* communication-act) containing the results ${}_jE_k^l$ (in RDF format). This procedure is repeated for each task's steps and eventually leads to either an agreement or an irreconcilable disagreement. Generally, the negotiation processes is a sequence of exchanged ACL messages; (both) parties use messages with *REQUEST* communication-act in order to ask for valid information and *INFORM* in order to reply. A detailed example that describes all the aspects concerning this integration is presented in the next section where a detailed use case scenario provides the whole process quite comprehensible.

## 5. USE CASE: A VIRTUAL TRAVEL AGENCY

Providing a quick and convenient way of exchanging goods and services, e-commerce has boomed. In the past decade, the vast majority of travel booking has moved online. Travel agents with virtual travel agencies were eventually well-positioned to take advantage of that shift in business. Even UNCTAD (UNCTAD 2012) proclaims that internet technology (ICT) and e-Commerce represent an opportunity for developing countries to improve their relative position in one of the most valuable aspects; tourism. In fact, agent technology has already been used in many applications that solve tourism issues, such as dynamic service discovery and vacation planning. A lot of researchers have studied the tourism field (Yueh et al. 2007; Gärtner et al. 2008; Moreno 2008; Zhang 2009; Park et al. 2012), agreeing that planning vacations and package tours is much easier using agent technology. Hence in this context, as e-Commerce applied extensively to the tour industry, we chose a virtual travel agency use case scenario to present the usefulness of agent technology in the field and the usability of our approach in particular.

Our use case scenario is loosely based on one of the most commonly used W3C's SWS usage scenarios (He et al. 2004). We adopt the main idea of a virtual travel agency that provides tourism services to end-users. Thus, we have a virtual travel agency, called VTA for short, which wants to offer to people-clients the ability to book complete vacation packages. These packages can cover all kinds of services concerned with tourism - from information about events and sights in an area to services that support booking of flights, hotels, rental cars, excursions, etc. Such VTAs are already existent, but at this point mostly comprise simple information portals along with some web-based customer services. By applying agent technology, a VTA will aggregate its services into a new automatic fashion. Such VTAs providing automated tourism services to end users thus tremendously enhance the functionality of currently existing VTAs. Finally, we mention that we deal only with agency – customer negotiation (B2C case), as the negotiation between the agency and other service providers, such as airline and credit card companies, is out of the scope of this study (B2B case).

Hence, this scenario involves two parties: a) the travel agency, represented by its agent, and b) a customer who wishes to buy a vacation package. In addition, there is an extra agent involved: c) the *Reasoner* an independent third-party service that provides reasoning services. In this use case scenario, a defeasible Reasoner was chosen since (nonmonotonic) defeasible logic is capable of modeling the way intelligent agents (like humans) draw reasonable conclusions from inconclusive information. At this point, a reasonable question would be why we need a Reasoner at all. The answer is simple; in this use case we tried to simulate a complex case in which agents do not share a common logic formalism in order to illustrate the usability of the approach even if the agents involved are unable to understand each other.

First of all, we adopt the use of a clause and a protocol (public) library, as already discussed. Additionally, both the VTA agent and the customer have their own private clause libraries containing their strategies. VTA is motivated by profit and thus it targets to customer satisfaction and eventually maximum packages sales, whereas customer wants to book vacation easily by choosing among a large variety of offers and targets to the best combination of services and prices for his/her needs. Hence, they can use these libraries in order to author and negotiate any potential e-Contract. According to our scenario, the customer wants to book a vacation package using a travel agency. First of all, he/she has to find the appropriate travel agency. For this purpose, he/she uses a directory service (*AYPS*), as described in the above section, where all travel agencies are registered. As a result, the customer finds an appropriate (VTA) agency. However, in order these agents (customer and VTA) to communicate and eventually negotiate they should either share a common formalism or use instead a third-party service (here called Reasoner). Hence, *AYPS* provides among others the logic formalism that is supported by the VTA. As a result, the customer is, now, able not only to communicate with the VTA but also to locate an appropriate Reasoner (by using again AYPS) that supports that formalism if he/she does not support it. To this end, a contract negotiation procedure between the customer and the VTA, that follows the aforementioned methodology, is able to begin.

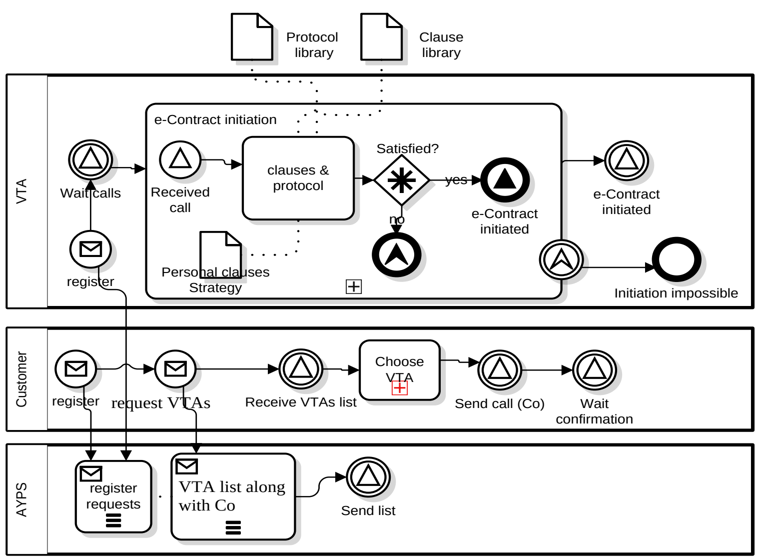


**Fig. 7.** Parties' registration and e-Contract authoring.

Therefore, the customer sends to VTA the requested call-for-negotiation ($C_0$) value, which in this case is the desired destination along with dates; Cuba and August in particular. The VTA agent searches the clause and protocol libraries along with its private clause repository and initiates an e-Contract (Fig. 7). Next, VTA invites the customer to establish trust and negotiate the e-Contract. Hence, both the VTA and the customer have to provide sufficient evidence in order to certify that they can trust each other. First of all, since it is forbidden to do business with people under the age of 18, the customer has to provide credentials (${}_{2}^{1}E_{2}^{1}$; the result of applying VTA's clauses ${}_{1}^{1}C_{1}^{1}$ to customer's data ${}_{2}^{1}D_{2}^{1}$) that certify his/her age (${}_{2}^{1}D_{2}^{1}$), letting the VTA to check it. Hence, at first an identity verification task is completed. Next, due to customer's preferences this VTA service has to be a member of a trusted third-party organization (VTA's clauses ${}_{2}^{1}C_{1}^{2}$), such as the *Better Business Bureau* (BBBO 2012), which guarantees that the service satisfies the standard criteria (membership verification task). Eventually, both reveal their credentials, successfully establishing trust among them (Fig. 8).

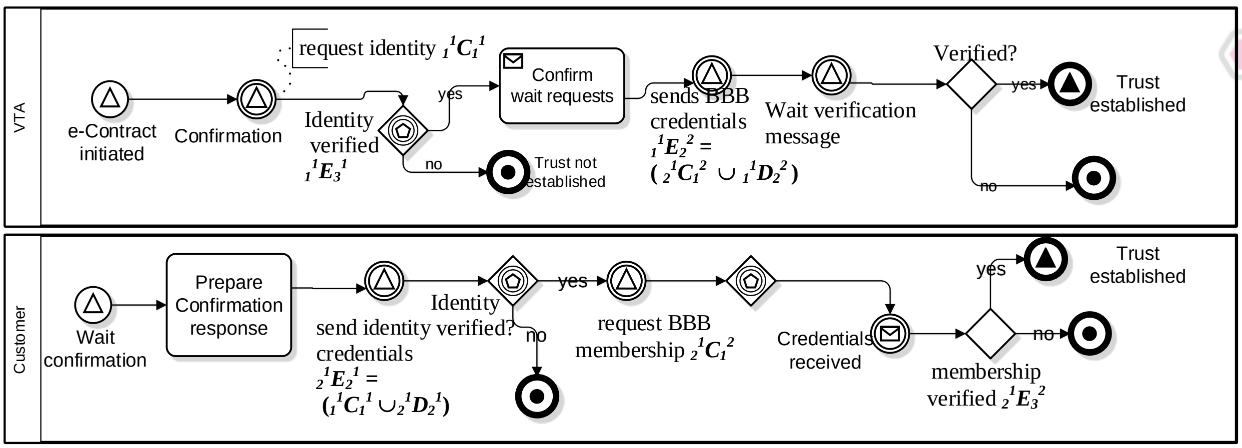


**Fig. 8.** Trust establishment.

Next there is the agenda setting and includes some primary data exchange, such as exchange of personal and credit data. To this end, the VTA agent wants to check if the customer is a returning one or not, as there are special offers for returning customers. Hence, the VTA agent sends a request to the customer, asking if he/she has a valid customer code (VTA's clauses ${}_{1}^{2}C_{1}^{1}$) but the customer replies negatively (${}_{2}^{2}E_{2}^{1}$; the result of applying VTA's clauses ${}_{1}^{2}C_{1}^{1}$ to customer's data). Next, during the next task, the VTA agent wants to know if the customer's desired destination and dates are flexible or not (VTA's clauses ${}_{1}^{2}C_{1}^{2}$) in order to recommend appropriate packages adjusting its negotiation restrictions. Once again the customer replies negatively; they are inflexible (${}_{2}^{2}E_{2}^{1}$; the result of applying VTA's clauses to customer's data). Finally, at the third and final task of this stage, the VTA agent sends part of the e-Contract's clauses (${}_{1}^{2}C_{1}^{3}$), in defeasible logic, requesting some of the customer's personal data. This request includes among others, the customer's name and e-mail, the credit card number and its date of expiration (Fig. 9). The customer is willing to reveal a part of its personal data, but internally uses a different type of logic and cannot directly process VTA's defeasible logic requirements.

Till then, it was able to answer VTA's plain requests but now it needs some help. To this end, an appropriate defeasible logic *Reasoner* is requested, which is retrieved from the directory service (*AYPS*), as already mentioned. This Reasoner uses the defeasible reasoning system DR-DEVICE (Bassiliades et al. 2006) in order to conduct inference. Hence, it accepts as input a defeasible logic rule base, written in the DR-RuleML language, which contains the rules and one or more RDF documents that contain the facts for the rule program, while the final conclusions are exported as an RDF document. Therefore, the customer communicates with the *Reasoner*, providing both its personal data (${}_{2}^{2}D_{2}^{3}$) in RDF and VTA's arguments (${}_{1}^{2}C_{1}^{3}$) in defeasible logic (RuleML format), and stands by for a reply. The *Reasoner* conducts inference on these arguments and data and produces the results as an RDF file. The Customer sends these results, namely its personal information that can be sent (${}_2^2E_2^3$), to the VTA agent.

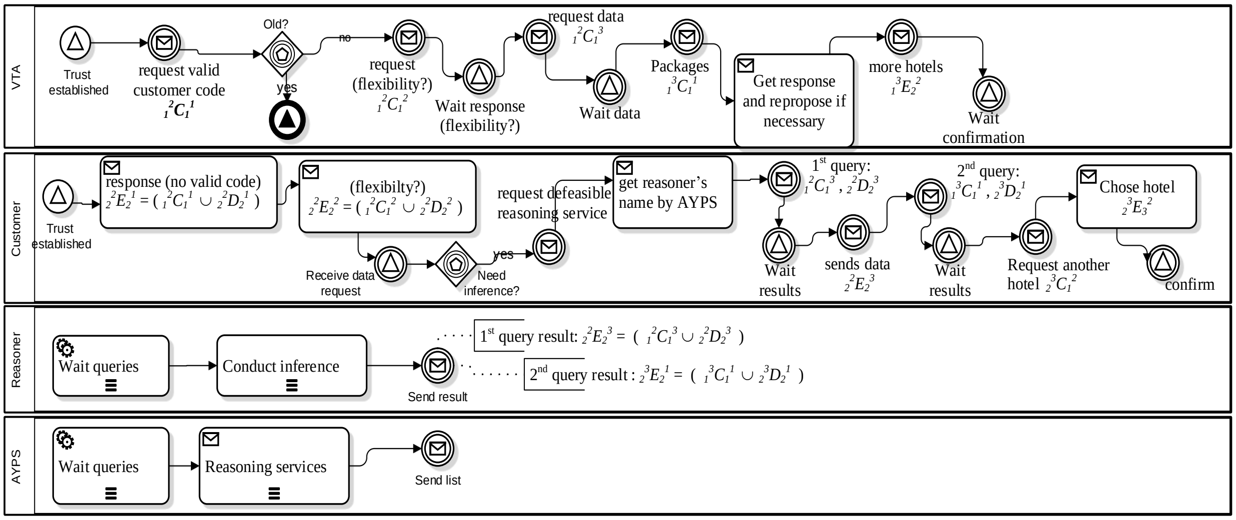


**Fig. 9.** Primary data exchange and hotel negotiation.

As soon as the VTA's agent receives the required data (${}_2^2E_2^3$), it proceeds to the first task (proposing packages) of the next stage by sending another part of the e-Contract (${}_1^3C_1^1$), a set of clauses represented in defeasible logic that contain the characteristics (some of them are marked as negotiable) of the three available vacation packages for Cuba in August (Table 3). The customer communicates again with the *Reasoner*, providing both the RuleML file containing these clauses (${}_1^3C_1^1$) and the RDF file which contains its preferences (${}_2^3D_2^1$); i.e. maximum price (1100€), minimum days (10), car for at least 5 days including car insurance, visa and map. Afterwards, the *Reasoner* sends the result (${}_2^3E_2^1$) where only the third package meets all its requirements however it is more expensive and thus rejected (Fig. 10 presents this package's general information in RDF format).

**Table 3.** Features of each vacation package.

| | | | |
|---|---|---|---|
| ***Package A*** | ***General info***<br>Destination= Cuba<br>Period=Summer<br>Duration=11 days/9 nights<br>Price = 859 ≤ 970€ | ***Included***<br>Hotel ≤ 3*<br>Airtickets→AirFrance,Aeroflot,IBERIA<br>Travel insurance (negotiable)<br>Bochure – map | ***Not included***<br>Visa = 25€ + 20 CUC<br>Airport taxies = 375€ |
| ***Package B*** | ***General info***<br>Destination= Cuba<br>Period=Summer<br>Duration=9 days/7 nights<br>Price = 940€ | ***Included***<br>Hotel 4*<br>Airtickets→AirFrance,Aeroflot,IBERIA<br>Rented car (up to 8 days) | ***Not included***<br>Visa = 25€ + 20 CUC<br>Airport taxies = 375€<br>Car insurance: negotiable |
| ***Package C*** | ***General info***<br>Destination= Cuba<br>Period=Summer<br>Duration=14days/12nights<br>Price = 1050 ≤ 1210€ | ***Included***<br>3* ≤ Hotel ≤ 5*:recommended 5*<br>Airtickets→AirFrance,Aeroflot,IBERIA<br>Rented car (up to 10 days)<br>Car insurance<br>Brochure – map, Visa | ***Not included***<br>Travel insurance<br>Airport taxies = 375€ |

The customer proceeds with the negotiation procedure (second task) in order to decrease the package's price. The customer lets the VTA to know that it is interested for the third package

(C) and wants to choose a 4* or 3* hotel instead of the package's 5* recommended hotel (${}_{2}^{3}C_{1}^{2}$; part of the customer's clauses) in order to reduce the price to 1100€ maximum. The VTA accepts and proposes new options (${}_{1}^{3}E_{2}^{2}$; the result of applying customer's clauses ${}_{2}^{3}C_{1}^{2}$ to VTA's data and clauses). The customer, being satisfied, chooses one of them. Next, the negotiation proceeds to the rest negotiation issues. Table 1 presented the ten most valuable issues, however in the context of this use case scenario change management and intellectual property are not eligible. Hence, the rest eight issues, presented in Fig. 3, are discussed among the VTA and the customer as above. However, we are not representing the procedure here in order to avoid repetition.


```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
xmlns:vta="http://www.vta.fake/vta#">

<rdf:Description rdf:about="http://www.vta.fake/vta/ Package C">
 <vta:destination>Cuba</vta:destination>
 <vta:period>Summer</vta:period>
 <vta:duration>
  <vta:days>14</vta:days>
  <vta:nights>12</vta:nights>
 </vta:duration>
 <vta:price>
  <vta:ValueFrom>1050</vta:ValueFrom>
  <vta:ValueTo>1210</vta:ValueTo>
 </vta:price>
</rdf:Description>
...
</rdf:RDF>
```


**Fig. 10.** VTA: Package C general information in RDF format.

Eventually, the VTA agent saves all the data concerning this customer and asks the final details (${}_{1}^{4}C_{1}^{1}$), e.g. the way of payment and the address/date that the customer wants to receive the e-tickets. The customer, on its behalf, accepts the deal and sends the final data (${}_{2}^{4}E_{2}^{1}$). Next, the e-Contract clauses are stored to the appropriate repository and both partners evaluate their strategy for future use. For example, the customer's evaluation value (EV – equation 1) for this use case is calculated as follows:

$$EV = \frac{CF_O}{CF_O + CF_G} \times OR + \frac{CF_G}{CF_O + CF_G} \times \frac{\sum_{i=1}^{9}(GF_i \times GR_i)}{\sum_{i=1}^{9} GF_i} = \frac{4}{4+6} \times 1 + \frac{6}{4+6}\,\frac{2\times1+4\times(-1)+3\times0+1\times1+4\times(-2)+1\times0+2\times1+2\times(-1)+1\times0}{20} = -0,05$$

The above equation reveals that the customer finds the satisfaction of the nine (eight of the ten proposed plus one for the hotel negotiation) negotiated goals ($CF_G$=6) more important than the final outcome ($CF_O$=4). Moreover, the customer evaluates also the importance ($CF_i$, $i \in [1,9]$) and his/her satisfaction ($GR_i$) for each of the nine goal as above ($CF_1$=2 x $GR_1$=1, $CF_2$=4 x $GR_2$=-1 and so on). The negative EV means that the customer is disappointed by the agreement. However, some of the nine goals where quite successfully negotiated. For example, the hotel negotiation was satisfactory (goal rating 1 – GR=1). The associated strategy in this case was new (R=0), hence its rating is now formed in R+=GR = 0+1=1 which means that it is likely to be used again in the future. Finally, throughout the e-Contract execution both parties proceed with their responsibilities agreed in the e-Contract while a performance evaluation (equation 2) could reveal how successful that transaction was.

### 5.1 Contract Terms and Information Specifications

Both the customer and the VTA's agent comply with EMERALD's KC-Agents model, thus, they are equipped with a Jess rule engine and a Jess knowledge base. Following the generic specification, presented in section 3, the VTA agent's description contains facts and rules. Facts, as already discussed, can be user-defined ($F^u$) or environment-asserted ($F^e$). For instance, user-defined fact *packages* represent part of VTA's internal knowledge and stands for the *packages characteristics* $F^u_{vta}$ = {*packages*}, whereas the environment-asserted fact *customer_name* stands for the customer's official name $F^e_{vta}$ = {*customer_name*}.

```
(defrule proposingPackages "propose vacation packages to the customer"
       (personal_info received) //stage 2 completed
       (MyAgent (name ?n))
       (customer ?x) (packages ?cc)
=>
(send (assert (ACLMessage (communicative-act REQUEST) (sender ?n) (receiver
?x)(content ?cc)))))
(defrule find_name
       (triple (subject ?x) (predicate rdf:type) (object sendable))
       (triple (subject ?x) (predicate CustomersName) (object ?name))
=>
(assert(name ?name)))
(defrule getChosenPackage "get the preferred package"
       (customer ?x)
       (customer_personal_info ?x ?p)
       ?z<-(ACLMessage(communicative-act INFORM)(sender ?x)(content ?c))
       (test (or (eq ?c A)(eq ?c B)(eq ?c C)))
=>
       (bind ?tt (new Basic))
       (bind ?str (?tt extractTriples ?p))
       (batch ?str)))
```

**Fig. 11.** Part of the VTA's behavior in Jess.

On the other hand, rules can lead to the execution of a special action, such as agent communication ($C_{vta}$) or Java call ($J_{vta}$). For instance:

$C_{vta}$ = {(*ACLMessage* (*communicative-act REQUEST*)(*sender VTA)*(*receiver* customer) (*content packages*)) ← *proposingPackages* (*"COND"*)}

stands for the rule "if it is time (*COND*) to propose available packages, send a message containing them to the customer". On the other hand:

$J_{vta}$ = {"*RDF"* ← ((*new java_class*) *getChosenPackage "COND*")}

stands for the rule "whenever it is necessary (*COND*), call the Java method *getChosenPackage* and get the available packages in RDF". To this end, Fig. 11 presents three of the VTA agent's behavior rules ($_1{}^3cnd_k{}^l$*; 1*$^{st}$ condition of the 3$^{rd}$ stage for the k$^{th}$ step of the l$^{th}$ task) as they were set by its user in Jess syntax. First, the *"find_name"* rule; after $stg_2$ is completed successfully finds the customer's name using the extracted triples in order to enter this customer in its registry. Next the "*proposingPackages"* rule that sends its clauses related to vacation packages to the customer and waits and finally the *"getChosenPackage"* rule that receives customer's reply, checks the selected package (A, B, C) and extracts the customer's personal data in RDF syntax in order to be able to provide them to the Reasoner (remember that the selected Reasoner requires data in RDF and rules in RuleML). Part of the aforementioned RDF data, required by the Reasoner in order to conduct inference and draw conclusions, was presented in Fig. 10.

Mention that the VTA's (${}_1{}^3C_1{}^1$) clauses that implement the vacation packages' characteristics (Table 3) are expressed in RuleML syntax. These clauses are the rules that must be combined with the (RDF) data and they are expressed in RuleML and not in Jess because they refer to publicly shared clauses as opposed to private behavior rules defined by the user in the Jess Knowledge Base. The customer based on its personal preferences and these clauses is able to select the most suitable package. Since these clauses are quite interesting and important we present them in a compact way by expressing them below in the d-POSL syntax of defeasible RuleML (Kontopoulos et al.2011). All three rules derive a positive conclusion, only one of which must be true, according to the constraint of the last line (conflicting literals).

```
r1: package-type(type→A):=
      pack-pref(Duration->11,Price->?price, Airtickets->Yes,Hotel->'3*',
                Travel_insurance->Yes, Rented_Car->No,
                Car_insurance->No,Car_days->0,Map->Yes,
                Visa->No,Airport_taxies->No),
      859≤?price, ?price≤970.
r2: package-type(type→B) :=
      pack-pref(Duration->9,Price->940,Airtickets->Yes, Hotel->'4*',
                Travel_insurance->No, Rented_Car->Yes,
                Car_insurance->No,Car_days->?cdays, Map->No,
                Visa->No,Airport_taxies->No),
      ?cdays ≤8.
r3: package-type(type→C) :=
      pack-pref(Duration->14,Price->?price,Airtickets->Yes,Hotel->?hotel,
                Travel_insurance->No, Rented_Car->Yes,Car_insurance->Yes,
                Car_days->?cdays, Map->Yes,Visa->Yes,Airport_taxies->No),
      1050≤?price, ?price≤1210,
      '3*'≤?hotel, ?hotel≤'5*',
      ?cdays≤10.
```

The relative "strength" of the rules, namely the superiority relation between them, is: `r3>r2>r1`.

Also, each package is unique ($Y \neq X$) and it actually defines a set of *conflicting literals*:

`C(package-type(X)) = {¬package-type(X)} ∪ { package-type(Y) | Y ≠ X }.`

Similarly, the customer's description contains, among others, a fact *personal_data* which is part of its internal knowledge, representing its personal data ($F^u_{customer} = \{personal\_data\}$), while due to the dynamic environment (AYPS is constantly updating the environment), new facts with the VTA's agent name (*VTA*) are added to the working memory ($F^e_{customer} = \{VTA\}$). Agent behavior is represented by rules, both for communication and Java method calls, that implement the workflow transition conditions of each negotiation step ${}^s cnd_k{}^l$ to the next.

## 6. RELATED WORK

A tightly related approach for automated agent negotiation is the *DR-NEGOTIATE* (Skylogiannis et al. 2007). It is a system based on the JADE framework that uses declarative negotiation strategies. Negotiation strategies are expressed in a declarative rule language, namely defeasible logic, and are applied using the implemented defeasible reasoning system DR-DEVICE (Bassiliades et al. 2006). A basic condition for the automation of the negotiation process in DR-NEGOTIATE is the existence of a negotiation protocol developed by the authors. This protocol is a finite state machine with discrete states and transition that must be hard-coded to all negotiation agents. Besides the fact that our approach proposes a workflow methodology for the overall process of an e-Contract agreement with evaluation management, even for the negotiation part itself, we propose a policy-based approach which is flexible, opposed to hard-coded protocols. Having a predefined hard-coded interaction protocol, like in DR-NEGOTIATE, requires significant programming time loss additionally to limiting the automation to a narrow amount of use cases. On the other hand, our policy-based approach is independent of the implementation language of the policy specifications, since any (agent) logic formalism can be applied. Hence, there is no need for hard-coded interaction protocols, any publicly available protocol can be used even if the involved parties use different implementation languages. However, we do agree in some issues. We admit the usefulness of defeasible logic (and the DR-DEVICE system) as we use it for demonstration and evaluation purposes in the provided use case scenario. Defeasible logic is capable of modeling the way intelligent agents (like humans) draw reasonable conclusions from inconclusive information, leading to more realistic conclusions and assessments similar to human reasoning and logic.

In (Silva et al. 2012) the authors present a web service negotiation process that considers human interaction and the use of different protocols for e-Contracts. This web service negotiation process is focused only on negotiation amongst organizations while our approach is more generic since it can be applied both in B2B and B2C e-Commerce and not only in business processes. Moreover, the authors promote the reuse of the core artefacts produced throughout the negotiation processes like our methodology which adopts the philosophy of clauses and policies reuse. Also, both approaches support different negotiation protocols. However, the authors support only web services hence they include interaction between human negotiators, on the other hand, our approach enables automatic negotiation procedure since intelligent agents act on behalf of their users maximizing their profit. Finally, only our approach takes into account the overall process of an e-Contract authoring to final approval including trust establishment among involved parties. Furthermore, regarding the negotiation phase itself only our approach takes into account issues related to proper negotiation term and goal use, automation and evaluation of the result and the used negotiation strategies.

Another related approach is *DR-CONTRACT* (Governatori and Pham 2009), an architecture for representing and reasoning on e-Contracts in defeasible logic also based on the DR-DEVICE architecture supplemented with a deontic defeasible logic of violation. The architecture captures the notions relevant to monitoring the execution and performance of e-Contracts in defeasible logics. More specifically, the framework deploys the Defeasible Deontic Logic of Violation (DDLV) (Governatori 2005), expressed via a RuleML extension that combines deontic notions with defeasibility and violations. DR-CONTRACT takes as input a DDLV theory, downloads/queries input RDF documents, including their schemata, and translates the RDF descriptions into fact objects. Finally, the conclusions are exported as an RDF/XML document through an RDF extractor. Once again, we do agree in usefulness of defeasible logic, although we do not use the deontic defeasible logic of violation. Moreover, our approach is not limited to the use of defeasible logic either. On the other hand, both approaches consider e-Contract mutual agreements between two or more parties engaging in various types of economic exchanges and transactions. However, DR-CONTRACT focuses on the monitoring of contract execution, omitting aspects of the process that leads to e-Contract agreement; i.e. trust establishment and terms negotiation. Our approach, on the other hand, focuses on the e-Contract agreement process itself and omits the after-agreement monitoring as it is not part of the negotiation and agreement process at all. In other words, our approach deals with the overall process of an e-Contract agreement whereas DR-CONTRACT deals with the reasoning and monitoring of an e-Contract's execution, based mainly on the implementation of an inference engine that supports defeasible deontic notions and violations.

*SweetDeal* (Grosof and Poon 2004) is another rule-based approach to representing contracts that enables software agents to create and execute contracts with substantial automation and modularity. SweetDeal builds upon the Situated Courteous Logic Programs (SCLP), represented in RuleML, which includes prioritized conflict handling and procedural attachments for actions and tests. Process knowledge descriptions are also incorporated, represented as ontologies expressed in DAML+OIL, thereby enabling more complex contracts with behavioral provisions, especially for handling exception conditions (e.g., late delivery or non-payment) that might arise during the execution of the contract. Hence, SweetDeal combines Semantic Web standards for knowledge representation of rules (RuleML) with ontologies (DAML+OIL/OWL). This provides a foundation for representing and automating deals about services – in particular, about Web Services. Similarly, our approach considers e-Contracts in the Semantic Web, but it is, to the best of our knowledge, the only one that provides a workflow methodology, which models the procedure that can be followed for negotiating and sealing an e-Contract. Our approach can be adopted in any multi-agent system even in the web services field. On the other hand, SweetDeal focuses on web services and ontologies, omitting multi-agent systems. Additionally, SweetDeal considers more the execution of an e-Contract and the exception conditions that might arise during the execution of the contract rather than the procedure that leads to the e-Contract agreement.

The *$ER^{EC}$ framework* (Krishna et al. 2004) is another example for designing and enacting e -Contract processes. The framework centers on an underlying meta-model that bridges the XML contract document with the Web Services-based implementation model of an e-Contract. The $ER^{EC}$ meta-model applies certain constructs for modeling e-Contracts, like clauses, activities and subcontracts. The framework also offers potential for automatic generating and deploying workflows for e-Contract enactment, as well as facilities for analyzing what-if scenarios with respect to e-Contract clause violation. Although, the $ER^{EC}$ framework and our approach focus on deferent directions, web services and mainly intelligent agents, respectively, they both consider e-Contracts in the Semantic Web as a complex procedure that has to be designed, modeled and executed properly. However, only our approach takes into account trust issues, an extremely important issue on on-line transactions and proposes a reusable policy-based workflow. Additionally, they also follow different directions for their implementation, although our methodology is system and technology independent since our implementation is just for demonstration and evaluation purposes whereas the methodology itself is a reusable implementation-independent procedure.

In (Lin 2008) the author presents a conceptual framework from the e-Contract representation perspective. This framework is one of wide-accepted conceptual models of the negotiation process for Web services contracting. It presents the overall process as a three entity collaboration, namely the service requester, the service provider and the service discovery agency. The contract manager, component of service requester, maintains a contract template for making agreements, which is actually not described at all. However, in the dynamic and flexible environment of e-Contract agreements, where so many heterogeneous parties interact, e-Contracts can never be static, rigid and agreed always under the same static contract template. Additionally, this approach does not clearly describe issues related to contract representation and term negotiation. Hence, opposed to our flexible, domain and system independent approach this framework seems more domain-dependent and static. Parties have no common understanding on the concepts they agree, an issue that is addressed in our approach with the use of trusted third-party reasoners. Moreover, the assumption that the service requester must maintain contract template is quite questionable. On the other hand, we propose an end-to-end solution with compliance to Semantic Web and e-Contracting standards.

In (Huang et al. 2010) the authors present a multiple-attribute negotiation model for B2C e-commerce, deploying intelligent agents to facilitate autonomous and automatic on-line buying and selling. The negotiation model includes a 4-phase model: information collection, search, negotiation, and evaluation. The authors also apply fuzzy theory and analytical hierarchy process to develop the system interface to facilitate the user inputs. In more detail, the agents prepare bids for and evaluate offers on behalf of the parties they represent with the aim of obtaining the maximum benefit for their users. Although, this model deals only with on-line buying and selling and not with e-Contracts in general, it is similar to our approach in the sense that it also considers negotiation as essential for B2C e-commerce systems and admits that multi-phases models is a promising solution for B2C future. Our approach, however, goes further by recognizing the need for a well-defined and reusable process not only in the negotiation phase but throughout the overall procedure of a B2C (or even a B2B) agreement among two or more parties. To this end, we take into account the parties policies in a flexible way as they are expressed through (private) policies and consider trust among parties essential. In other words, our approach is human thinking oriented performed automatically and independently in any multi-agent system, providing a safe and reusable management framework.

Concerning commitment-based contracts, a representative case was presented in (Udupi and Singh 2006). The authors proposed a commitment-based architecture for agent-based virtual organizations. Each virtual organization is associated with a set of goals, commitments, and policies while its members behave independently, constrained only by their contracts. The proposed study provides a commitment-based architecture for contract modeling and enactment which is able to handle various scenarios of conflicts and exceptions in contract enactment. Based on that architecture, contracts may be formed at multiple levels. This approach is similar to ours in the sense that both approaches consider goals and policies core parts of any multi-level contract (hierarchical subcontracts in our case). Hence, although our approach does not include commitments while Udupi and Singh’s (2006) approach does not include workflows, both approaches have a similar point of view regarding contract modeling and execution. Yet, our approach focuses on three important issues, namely trust establishment, term negotiation and evaluation (both strategy and performance), proposing methods that can address the challenges related to these issues. Furthermore, taking into account that agents usually do not understand each other’s formalisms, we propose the integration of our methodology into a knowledge-based system that provides, among others, trusted third-party reasoning services to resolve logic formalism heterogeneity.

Singh and his partners studied commitments and contracts in a number of cases, such as in (Chopra et al. 2011). In that study the authors proposed an approach to assessing the robustness of contracts. They proposed a methodology for creating commitment-based models of contracts from textual descriptions. Additionally, they studied how contracts can be evaluated for robustness. Although, we do consider robustness of contracts important, we investigate the issue under a different perspective, including trust establishment and mainstream goal negotiation in our workflow methodology. Additionally, we focus more on agent's profit in the sense that we try to assist them improving their behavior through strategy and performance evaluations.

Concerning norm-based contracts, an important case was the IST-CONTRACT project (CONTRACT 2006). The project's work was documented, among others, in (Miles et al. 2008). In that representative article, the authors presented the CONTRACT conceptual framework and architecture, where a contract is defined as a set of clauses, each of which can be viewed as a normative statement, or norm. The authors discussed how the contractual obligations imply critical states of a system, namely states that an application may reach, allowing contracts to be checked for conflicts and violations. Furthermore, they identified four key process types in the contract lifecycle, namely establishment, maintenance and update, fulfilment and termination or renewal. Yet, the authors claimed that these process types could be differentiated from application to application. Our approach, similar to CONTRACT, divides the lifecycle of an e-Contract into stages, process types in the phraseology of CONTRACT. Yet, we specify five strict stages instead of four flexible since we focus on a well-defined workflow methodology. One of our main differences is found in the negotiation phase. This phase is omitted in the above approach, which focus on obligations and violations, whereas it is a core issue in ours, where we study ten negotiation mainstream goals that can improve parties' profit. Furthermore, our approach includes and studies trust issues related to the involved parties. Finally, our approach promotes not only e-Contract's efficiency but also parties' behavior importance by providing metrics for performance and strategy evaluation.

## 7. CONCLUSIONS AND FUTURE WORK

The article discussed about e-Commerce, which although met a massive growth over the past years is still rather complicated, due to the fact that the parties involved have to collect information, negotiate contracts and safely execute transactions. Eventually, the involved parties have to reach an agreement, namely an e-Contract for each successful e-Commerce transaction. So far, there are plenty of approaches that deal partially with this subject. This study focused on how the end-to-end procedure could be fully integrated, automated and system independent. The complication and automation problem was addressed by using intelligent agents acting in the Semantic Web, as agents can perform the same tasks unsupervised while Semantic Web provides interoperability and realistic information organization in a way similar to human mind. Yet, in order to make agents reach their maximum efficiency, a well – formed modeling of the transaction process is needed. Hence, this article presented a reusable policy-based workflow methodology for efficient B2C (or even B2B with minor implementation extensions) contract agreement among agents, where each agent has its private policies and strategies simulating human thinking and acting. It deals with the overall process of an e-Contract agreement from the trust establishment among the parties to the final agreement and approval.

Next we studied the evaluation process, another core issue in e-Contract management, where we tried not only to manage an e-Contract's performance but also to use this knowledge in the negotiation process. Our approach deals with strategy and performance evaluation enabling agents evaluate both their own and the rest parties' behavior. The founding of this evaluation is used in order to gradually improve parties acting, gaining greater benefits from their trade. Finally, we tried to figure out how our approach could adopt really promising solutions in the rest of the lifecycle in order to provide a stable and reusable end-to-end management proposal.

In addition, the integration of this methodology into a multi-agent knowledge-based framework, called EMERALD, was proposed and demonstrated, providing flexibility, reusability and interoperability of behavior between agents. EMERALD provides a number of reasoning services for a variety of logic and reasoning formalisms letting the agents understand their partners' different formalisms. Hence, the main advantage of this integration is that it provides a safe and reusable framework for e-Contract management. Finally, an e-Commerce contract negotiation scenario was presented that demonstrated the added value of the approach. Throughout our proposed methodology and its demonstrated case study, we focused on using Semantic Web technologies for data and policy exchanges, namely RDF and RuleML respectively, in order to maximize interoperability among the involved parties (systems). Even in the case of rules exchange, where interoperability is not enforced by simply using a common syntax, we have utilized a multi-agent interoperability framework (EMERALD).

So far, we have focused our efforts on B2C e-Contracts; in the future, we plan to study B2B e-Contracts agreements in order to evaluate our approach. However, our main future direction is to extend the proposed methodology in order to capture the wider notion of trust in the Semantic Web. More specific, currently, researchers, motivated by the fact that the degree of trust that can be invested in a certain agent is recognized as a vital issue, focus their efforts on the last two Semantic Web layers, namely trust and proof layer. We have so far, studied and proposed a variety of trust models (Kravari et al. 2010d; Kravari and Bassiliades 2012) and our intention is to adopt them in the principles of methodology presented in this paper.

More specific, currently the trust establishment stage is based on providing the essential amount of certificates, verifying ID or membership, since e-Contract procedures are quite isolated from the latest research on trust in MASs. Hence, we plan to bring these two areas close by integrating the trust layer's accomplishments in e-Contract procedures. Additionally, we plan to go a step further by studying and eventually integrating the proof layer's principles, too. Transactions among parties in the future will contain proof checking mechanisms (e.g. Kravari et al. 2011; Bassiliades et al. 2007). Hence, our intention is to extend the methodology in order to integrate the generated proofs with the trust mechanisms, in order to interconnect the proof and trust layers of the Semantic Web in the e-Contract procedures.

## REFERENCES

Antoniou G, Dimaresis N, Governatori G (2009) A modal and deontic defeasible reasoning system for modelling policies and multi-agent systems. *Expert Systems with Applications, 36*(2), 4125-4134

Antoniou G, Skylogiannis T, Bikakis A, Doerr M, Bassiliades N (2007) DR-BROKERING: A Semantic Brokering System. *Knowledge-Based Systems*, *20*(1), 61-72

ASC (2013) Advanced Software Concepts. United States: 16 Corporate Way, Suite 2-3284, Valley Cottage, NY 10989, USA

Bassiliades N, Antoniou G, Governatori G (2007) Proof Explanation in the DR-DEVICE System. *In proceeding of the 1th Int. Conf. on Web Reasoning and Rule Systems (RR '07)*, LNCS 4524, pp. 249-258

Bassiliades N, Antoniou G, Vlahavas I (2006) A Defeasible Logic Reasoner for the Semantic Web. *International Journal on Semantic Web and Information Systems, 2*(1):1-41

*BBBO: Better Business Bureau Organization* (2012) Retrieved November 29, 2012 from http://www.bbb.org/

Bellifemine F, Caire G, Poggi A, Rimassa G (2003) JADE: A white Paper. *EXP in search of innovation, 3*(3), 6-19

Berners-Lee T, Hendler J, Lassila O (2001) The Semantic Web. *Scientific American Magazine,284*(5), 34-43 (Revised 2008)

Boley H, Kifer M (2010) A Guide to the Basic Logic Dialect for Rule Interchange on the Web. *IEEE Transactions on Knowledge and Data Engineering, 22*(11), 1593-1608

Boley H, Paschke A, Shafiq O (2010) RuleML 1.0: The Overarching Specification of Web Rules. *In proceeding of the 4th Int. Web Rule Symposium: Research Based and Industry Focused (RuleML'10)*, vl.6403, pp. 162-178

*BPMN, Object Management Group: Business Process Model and Notation* (2012) Retrieved November 29, 2012, from http://www.bpmn.org/

Chiu D, Cheung S, Till S, Karlapalem K, Li Q, Kafeza E (2004) Workflow view driven cross-organizational interoperability in a web service environment. *Information Technology and Management*, *5*(3-4), 221-250

Chopra A, Oren N, Modgil S, Desai N, Miles S, Luck M, Singh M (2011) Analyzing Contract Robustness through a Model of Commitments. Agent-Oriented Software Engineering XI, LNCS Vol. 6788, pp. 17-36.

CONTRACT (2006) Contract based Systems Engineering Methods Contract based Systems Engineering Methods for Verifiable Cross for Verifiable Cross -Organisational Networked Organisational Networked Business Applications Business Applications, 6th Framework Programme, pr. number FP6-034418.

Daskalopulu A, Dimitrakos T, Maibaum T (2001) E-Contract fulfilment and agents' attitudes. Presented in ERCIM WG E-Commerce Workshop on The Role of Trust in e Business, Zurich

Degel M (2013) *Adopting Semantic Technology in the Enterprise*. Opening keynote at Semantic Technology & Business Conference (SemTechBiz), San Francisco, USA

Dunne P, Wooldridge M, Laurence M (2005) The complexity of contract negotiation. Artificial Intelligence, 164(1–2):23-46

Exari Systems (2013) Exari Document Assembly and Contract Management. Melbourne: Level 7, 10-16 Queen Street, Melbourne 3000, Australia

Gärtner M, Seidel I, Berger H (2008) Agent mediated trading in a 3D e-tourism environment. *In Proceedings of the 10th international conference on Electronic commerce (ICEC '08)*, pp. 14:1--14:10

Gattani A (2013) *How to Connect People and Products Using Big, Fast, and Heterogeneous Data*. Opening keynote at Semantic Technology & Business Conference (SemTechBiz), San Francisco, USA

GoodRelations Project (2013) The Web Vocabulary for E-Commerce. Universität der Bundeswehr München. E-Business and Web Science Research Group, Germany

Governatori G (2005) Representing Business Contracts in RuleML. *International Journal of Cooperative Information Systems, 14*(2-3, 181–216

Governatori G, Dumas M, Hofstedeter A, Oaks P (2001) A Formal Approach to Protocols and Strategies for (Legal) Negotiation. ICAIL 2001, pp. 168-177

Governatori G, Pham DH (2009) DR-CONTRACT: An Architecture for e-Contracts in Defeasible Logic. *International Journal of Business Process Integration and Management, 4*(3), 187-199

Governatori G, Rotolo A (2008) BIO Logical Agents: Norms, Beliefs, Intentions in Defea-sible Logic. *Journal of Autonomous Agents and Multi_Agent Systems, 17*(1), 36-69

Grosof BN, Poon TC (2004) SweetDeal: Representing Agent Contracts With Exceptions using Semantic Web Rules, Ontologies, and Process Descriptions. *International Journal of Electronic Commerce (IJEC), 8*(4):61-98, special issue on web e-commerce

Gummesson E, Polese F (2009*)* B2B is not an island!. *Journal of Business & Industrial Marketing*, 24(5/6):337 – 350

He H, Haas H, Orchard D (2004) Web Services Architecture Usage Scenarios. W3C Working Group Note, 11 February 2004. Retrieved November 1, 2012, http://www.w3.org/TR/ws-arch-scenarios/

Hendler J (2001) Agents and the Semantic Web. *IEEE Intelligent Systems, 16*(2), 30-37

Hensher D, Stanley J (2008) Transacting under a performance-based contract: The role of negotiation and competitive tendering. *Transportation Research Part A: Policy and Practice*, 42(9):1143-1151

Huang CC, Liang WY, Lai YH, Lin YC (2010) The agent-based negotiation process for B2C e-commerce. *Expert Systems with Applications, 37*(1), 348-359

Huang H, Kauffman R, Xu H, Zhao L (2011) Mechanism design for e-procurement auctions: On the efficacy of post-auction negotiation and quality effort incentives. *Electronic Commerce Research and Applications, 10*(6), 650-672

Hutt M, Speh T (2012) *Business Marketing Management: B2B (11th ed.)*. Mason, USA: South-Western, Cengage Learning

*IACCM: The International Association for Contract & Commercial Management.* Worldwide Head Office, 90 Grove Street, Ridgefield, CT 06877, USA

Indukuri KV, Krishna PR (2010) Mining e-contract documents to classify clauses. *In Proc. of the Third Annual ACM Bangalore Conference (COMPUTE '10)*, Article 7, 5 pages

Infosys (2013) White Paper: Contract Lifecycle Management. The DNA of Procurement. Retrieved January 15, 2014, from http://www.infosys.com/supply-chain/white-papers/Documents/contract-lifecycle-management.pdf

Jertila A, Schoop M (2005) LAP and Semantic Web: A Language Action Perspective on Electronic Contracts. *In proceedings of the 10th International Working Conference on the Language Action Perspective on Communication Modeling* (LAP 2005), pp. 157-171

*JESS, the Rule Engine for the Java Platform* (2008) Retrieved November 29, 2012, from http://www.jessrules.com/

Karlapalem K, Krishna R (2006) State-of-the-Art in Modeling and Deployment of Electronic Contracts. *Advances in Conceptual Modeling - Theory and Practice*, LNCS, Volume 4231, pp 3-4. Berlin Heidelberg: Springer-Verlag

Kontopoulos E, Bassiliades N, Antoniou G (2011) Visualizing Semantic Web proofs of defeasible logic in the DR-DEVICE system. *Knowledge-Based Systems, 24*(3), 406-419

Koons R (2009) *Defeasible Reasoning*. Stanford Encylopedia of Philosophy. Stanford University. Retrieved November 29, 2012, http://plato.stanford.edu/entries/reasoning-defeasible/

Koppensteiner G, Merdan M, Lepuschitz W, Moser T, Reinprecht C (2011) Multi Agent Systems combined with Semantic Technologies for Automated Negotiation in Virtual Enterprises. *Modeling, Control, Programming, Simulations and Applications*, INTECH, pp. 221 - 240

Kravari K, Bassiliades N (2012) Advanced Agent Discovery Services. In proc. of the 2nd International Conference on Web Intelligence, Mining and Semantics (WIMS '12), Article 38, pp. 338-349

Kravari K, Kastori GE, Bassiliades N, Governatori G (2010) A contract agreement policy-based workflow methodology for agents interacting in the semantic web. *In Semantic Web Rules, Proceedings of the 4th International Web Rule Symposium (RuleML 2010)*: LNCS, Vol. 6403, 225 – 239 (Best Paper Award)

Kravari K, Kontopoulos E, Bassiliades N (2009a) A Trusted Defeasible Reasoning Service for Brokering Agents in the Semantic Web. *In proceedings of the 3rd International Symposium on Intelligent Distributed Computing (IDC'09)*,Vol. 237, pp. 243-248

Kravari K, Kontopoulos E, Bassiliades N (2009b) Towards a Knowledge-based Framework for Agents Interacting in the Semantic Web. *In IEEE/WIC/ACM Int. Conf. on Intelligent Agent Technology (IAT'09)*, Vol. 2, pp. 482-485

Kravari K, Kontopoulos E, Bassiliades N (2010a) EMERALD: A multi-agent system for knowledge-based reasoning interoperability in the semantic web. In S. Konstantopoulos, S. J. Perantonis, V. Karkaletsis, C. D. Spyropoulos, & G. A. Vouros (Ed.), Artificial Intelligence: Theories, Models and Applications, 6th Hellenic Conference on Artificial Intelligence, SETN 2010: LNCS, Vol. 6040/2010, 173-182

Kravari K, Kontopoulos E, Bassiliades N (2010b) Trusted Reasoning Services for Semantic Web Agents. *Informatica: International Journal of Computing and Informatics, 34*(4), 429-440

Kravari K, Malliarakis C, Bassiliades N (2010) T-REX: A hybrid agent trust model based on witness reputation and personal experience. *In Proceeding of 11th Int. Conf. on Electronic Commerce and Web Technologies (EC-Web 2010): LNBIP, 61*(3), 107-118

Kravari K, Papatheodorou K, Antoniou G, Bassiliades N (2011). Reasoning and Proofing Services for Semantic Web Agents. *In proceedings of the 22nd Int. Join Conf. on Artificial Intelligence (IJCAI-2011)*. Best Paper Track, Vol.3, pp. 2662-2667

Krishna PR, Karlapalem K (2008) Electronic contracts. *IEEE Internet Computing*, *12*(4), 60-68

Krishna PR, Karlapalem K, Chiu DK (2004) An EREC Framework for e-contract Modeling, Enactment and Monitoring. *Data Knowledge Engineering*, *51*(1), 31-58

Krishna PR, Karlapalem K, Dani AR (2005) From contracts to e-contracts: Modeling and enactment. *Information Technology and Management*, *6*(4), 363-387

Laudon K, Traver CG (2012) *E-Commerce 2012* ($8^{th}$ ed.). Upper Saddle River, New Jersey: Prentice Hall

Lee A, Seamons K, Winslett M, Yu T (2007) Automated Trust Negotiation in Open Systems. *Secure Data Management in Decentralized Systems*, *Advances in Information Security*: Volume 33, 217-258, Springer

Lin J (2008) A conceptual model for negotiating in service-oriented environments. *Information Processing Letters*, 108(4):192–203

Maher MJ (2001) Propositional defeasible logic has linear complexity. *Theory and Practice of Logic Programming*, 1(6):691–711

Mawji A (2005) Contract Life-Cycle Management. Retrieved January 15, 2014, from http://www.ncmahq.org/files/Articles/390E6_CM_Nov05_p38.pdf

Miles S, Oren N, Luck M, Modgil S, Faci N, Holt C, Vickers G (2008). Modelling and Administration of Contract-Based Systems. *In Proceedings of the AISB 2008 Symposium on Behaviour Regulation in Multi-Agent Systems*, pp. 19-24

Moreno A (2008) Agent Applications in Tourism. *Issues in Multi-Agent Systems, Whitestein Series in Software Agent Technologies*, pp.179-206

*NCMA: National Contract Management Association. Headquarters:* NCMA, 21740 Beaumeade Circle, Suite 125, Ashburn, Virginia 20147, USA

Nute D (1987) Defeasible Reasoning. *In proceedings of the 20th International Conference on Systems Science*, pp. 470-477

Park H, Yoon A, Kwon HC (2012) Task Model and Task Ontology for Intelligent Tourist Information Service. *International Journal of u- and e- Service, Science and Technology*, Vol. 5, No. 2, pp. 43-58

Patterson M (2010) Standardization of Standard-Form Contracts: Competition and Contract Implications. *William and Mary Law Review*, 52(2):327-414

Pollock JL (1992) How to reason defeasibly. *Artificial Intelligence, 57*(1), 1-42

RDF Specifications: Resource Description Framework (RDF) Model and Syntax Specification (2004) Retrieved November 29, 2012, from http://www.w3.org/TR/PR-rdf-syntax/

Šaučiūnas M (2012) Automated eContract Negotiation in Web Service Environment: Electronic Contract Management Aspects. *Tenth International Baltic Conference on Databases and Information Systems*, pp. 241-247

Silva GC, de Souza Gimenes IM, Fantinato M, de Toledo BF (2012) Towards a Process for Negotiation of E-contracts Involving Web Services. *VIII Simpósio Brasileiro de Sistemas de Informação* (SBSI 2012), pp. 267-278

Skylogiannis T, Antoniou G, Bassiliades N, Governatori G, Bikakis A (2007) DR-NEGOTIATE – A System for Automated Agent Negotiation with Defeasible Logic-Based Strategies. *Data & Knowledge Engineering, 63*(2), 362-380

Stadermann J, Symons S, Thon I (2013) Extracting hierarchical data points and tables from scanned contracts. *UIMA@GSCL*, volume 1038, pp, 50-57

Stuart R, Norvig P (2009) *Artificial Intelligence: A Modern Approach* (3nd ed.). Upper Saddle River, New Jersey: Prentice Hall

Turban E, King D (2011) *Electronic Commerce 2012: Managerial and Social Networks Perspectives* (1st ed.). Upper Saddle River, New Jersey: Prentice Hall

Udupi Y, Singh M (2006) Contract Enactment in Virtual Organizations: A Commitment-Based Approach. Proceedings of the 21st National Conference on Artificial Intelligence (AAAI), pp. 722-728

*UNCTAD: United Nations Conference on Trade and Development 2012*. Retrieved November 29, 2012, from http://www.unctad.org

Vidyasankar K, Krishna R, Karlapalem K (2007) A Multi-level Model for Activity Commitments in E-contracts. On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS, LNCS Vol. 4803, pp. 300-317

Yueh YTF, Chiu DKW, Leung H, Hung PCK (2007) A Virtual Travel Agent System for M-Tourism with Semantic Web Service Based Design and Implementation. *Advanced Information Networking and Applications*, AINA '07, pp.142-149

Zhang XM (2009) A Semantic Grid Oriented to E-Tourism. *CloudCom 2009*, LNCS 5931, pp. 485-496

Zycus (2013) 5 Steps to Realizing Value from Contract Lifecycle Management. Retrieved January 15, 2014, from http://www.zycus.com/newsletter/jan-2013/5step-contract-lifecycle-management_final.pdf