# A rule-based eCommerce methodology for the IoT using trustworthy Intelligent Agents and Microservices

Kalliopi Kravari✉[0000-0002-2298-9038] and Nick Bassiliades[0000-0001-6035-1038]

Dept. of Informatics, Aristotle University of Thessaloniki, Thessaloniki GR-54124, Greece
✉kkravari@csd.auth.gr
nbassili@csd.auth.gr

**Abstract.** The impact of the Internet of Things will transform business and economy. This network of intercommunicating heterogeneous Things is expected to affect the commerce industry by driving innovation and new opportunities in the future. Yet, this open, distributed and heterogeneous environment raises challenges. Old eCommerce practices cannot be sufficiently applied while trustworthiness issues arise. This study proposes a rule-based eCommerce methodology that will allow Things to safely trade on the network. The proposed methodology represents Things as Intelligent Agents since they form an alternative to traditional interactions among people and objects while they are involved in a rich research effort regarding trust management. It also combines Intelligent Agents with the microservice architecture in order to deal with Things heterogeneity while it adopts the use of a social agent-based trust model. Well-known semantic technologies such as RuleML and defeasible logic is adopted in order to maximize interoperability. Furthermore, in order to deal with issues related to rule exchange with no common syntax, the methodology is integrated to a multi-agent knowledge-based framework. Finally, an eCommerce scenario is presented, illustrating the viability of the approach.

1. **Keywords:** Multi-agent Systems, Defeasible Reasoning, Trustworthiness.

## 1 Introduction

The Internet of Things (IoT) seems to be an emerging IT technology. Its main innovation consists of creating a world where Things, devices, services or even humans, will be connected and able to make decisions and communicate [7]. An area that is expected to attract attention is eCommerce which has achieved a growth but due to IoT emergence, it faces new challenges. It must be clearly recognized that the application of IoT is still at an early stage while the relevant technology is not mature. Today, the IoT mostly sends data up towards the Cloud for processing. Many researchers believe that as both software and hardware continues to evolve, some of these processes may be bring back to the devices. Hence, in the IoT of tomorrow, value between devices and across industries could be uncovered using Intelligent Agents (IAs) that can add autonomy, context awareness, and intelligence [7]. Besides, current eCommerce

evolved on the basis of the past retail sector, hence, product quality is difficult to guarantee, the pay security, logistics and distribution systems need more automation. As more devices get connected and gain smart features, more data will be gathered, and consumer experience can be improved. Hence, although all these requirements exists in the IoT eCommerce, they have to be upgraded with intelligence, autonomy and semantic awareness (despite the need semantic languages are not used yet). With the penetration of IoT, a better management of inventory, easy loss tracking, increase in shopper intelligence and intelligent logistics systems with timeliness, convenience and safety properties should be developed. However, deployment of IoT to facilitate eCommerce applications raises important challenges such as information exchange and trust issues [3, 7]. In this context, we, as plenty others, propose a decentralized approach where devices combined with agents will become part of the Internet of Smart Things. Of course, there is much work to be done regarding decentralized Multiagent systems as a way to place decentralized intelligence in distributed computing.

The aim of this study is to propose a rule-based eCommerce methodology that will allow Things to locate proper partners, establish trust relationships and interact in the future IoT network. The proposed methodology adopts the use of IAs that are considered a technology that can deal with these challenges while there are involved in a rich research effort regarding trust management, even though it refers to Semantic Web [3]. The methodology integrates an IoT social agent-based reputation model in order to allow trust establishment in the environment. Furthermore, this study combines the agent technology with the microservice architecture, a promising modular approach [1]. A core concept of the methodology is the proper information exchange among Things in order to assure safe and robust transactions. Hence, the base of the methodology is a weakly-structured workflow with strong information technologies. Hence, well-known semantic technologies such as RuleML and defeasible logic is adopted, maximizing interoperability, reusability, automation and efficiency among Things. Additionally, the methodology is integrated to a multi-agent knowledge-based framework that supports rule exchange without the need of a common syntax.

## 2 Rule-based eCommerce methodology

All Things are represented as agents while microservice architecture is used for the implementation of services and devices [3]. Microservices have almost every known agent property while they allow control over even hardly reached devices. Here, we propose two types of microservices called DMicro, related to an IoT device, and SMicro, related to a service, represented as agents. Each IoT-oriented eCommerce website should have at least a SMicro agent that will act as proxy between the site and the Things. Hence, the proposed methodology involves three types of agents, modeling IoT Things, called $T_{hv}$ (humans/virtual), $T_{DMicro}$ (devices) and $T_{SMicro}$ (services). Each agent's conceptual base follows this $T_x$ tuple specification: A = {ID, $T_x$, LC, LP, B, S, ER, EV}(1), where ID is the agent's name, $T_x$ is its type $\{x \equiv T_{hv} | T_{SMicro} | T_{SMicro}\}$, LC and LP are extendable lists of characteristics C and preferences P $\{LC_x^n \,\&\, LP_x^n \mid n \in [1, N], x \equiv agent\}$, B is agent's beliefs (facts) $\{B = B_I \cup B_R \mid$ $B_I$ by agent's interactions,

$B_R$ by reasoning processes}, S is its strategy rules, ER is the ratings derived by the agent's direct experience and EV contains its evaluation criteria. The LC list includes information such as the type of provided products/services. Preferences (LP) include information such as the desirable delivery time. For computational and priority purposes, each characteristic and preference is optionally assigned with a value of importance (weight) at the range [0, 1] ($W_c^n$ & $W_p^n$ | $n \in [0, 1]$, $c \equiv$ *characteristic*, $p \equiv$ *preference*), defining how much attention will be paid to it.

In this context, we propose a five stage weakly-structured workflow methodology for the IoT-oriented eCommerce, based on the philosophy that the procedure can be divided into groups of tasks forming special categories containing a number of steps. The proposed approach acknowledges five stages; 1. *Locate potential partners*, 2. *Establish trust*, 3. *Exchange request/data*, 4. *Negotiate terms*, 5. *Reach agreement*. Here, we focus mainly on the two first stages, namely locating partners and establish trust since they are critical for IoT success, yet, they are not sufficiently studied. In general, stages represent the main issues of the overall eCommerce process while steps represent individual actions that agents have to handle depending on their rule-based strategy. Thus, each specific stage requires a group of steps. Transitions between stages are sequential, but transitions between steps may be not. The involved agents have to proceed gradually from the first issue (locate partners) to the last while steps represent individual actions referring even to optional or repeatable actions.

In order to present the terms and defeasible rules of the approach in a compact way, we express them in the compact d-POSL syntax [4] of defeasible RuleML. Defeasible logic (DL) has the notion of rules that can be defeated. In DL strict rules (B:-A1, …, An in d-POSL syntax) are rules in the classical sense. Defeasible rules (B:=A1, …, An) can be defeated by contrary evidence. Defeaters (B:~A1, …, An) are a special kind of rules, used to prevent conclusions and not to support them. DL does not support contradictory conclusions, instead seeks to resolve conflicts. In cases where no conclusion can be derived, a priority is expressed through a superiority relation among rules which defines where one rule override the conclusion of another.

*Locating potential partners.*

The proposed approach offers two alternative options for locating partners, one of the major challenges for open distributed and large-scale systems. The first one is based on auction-inspired broadcasting with no rating requirements while the second is based on LOCATOR [24] which requires previously reported reputation ratings.

*Broadcasting option*: Whenever an agent (Tx) has no previous interaction history or it needs fast SMicro locating, it broadcasts a CFP reaching easily available agents. Agents that receive the CFP call either ignore it or reply with a propose message.

$r_{i1}$: *broadcastCFP*(*id*→*?id*$_x$, *time*→*?t*, *sender*→*?x*, *receiver*→*?SMicro*, *typeRequest*→*?typeRequest*, *reasonImportance*→*?rim*, *specs*→*?specs*) :- *timeAvailablity(low)*,*request*(*typeRequest*→*?typeRequest*$_x$,*specs*→*?specs*$_x$,*reasonImportance*→*?rim*$_x$), *reasonImportance_threshold*(*?rim*), *?rim*$_x$>*?rim*, *?typeRequest*$_x$ = *?typeRequest*.

A typical CFP message ($r_{i1}$) indicates that the sender asks the receiver (*?SMicro*) about a specific reason (*typeRequest*, e.g. rating request), stating how important is that request (*reasonImportance* at the range [0, 1]) and which are the specifications of the

request (*specs, e.g.* service type). It will send the CFP if the time availability is low (there is such a fact to its beliefs: $B_R$={*timeAvailablity(low)*}), the importance of the reason is greater than its threshold (*reasonImportance_threshold*(*?rim*), $?rim_x > ?rim$) and it needs to fulfil such a type request (holds such goal: G={fulfil($?typeRequest_x$)}).

*LOCATOR variation option*: The second option is based on the philosophy of LOCATOR, a locating rating mechanism that uses features from both social graphs and peer-to-peer networks in order to incorporate potential social dimensions and relationships within the IoT. Here, we adopt the use of LOCATOR to locate potential SMicro partners. Our variation works as follow: A $T_x$ agent interested in a $T_{SMicro}$ agent, based on its preferences ($LP_{TR}^n$) decides upon the characteristics ($LC_{TE}^n$) it considers important. It assigns proper weights ($W_c^n$) to them and searches its database for previously known agents that fulfil its requirements in order to ask them. Characteristics that weight more are more important in the sense that $T_x$ believes that partners with these characteristics will be more reliable (social influence). In this context, $T_x$ depending on its personal strategy ($s_x^n \in n\ [1, N], x \equiv agent$) sends an offer request to known agents with one or two high-weighted characteristics. If the feedback is not satisfying, it sends requests to partners with lower-weighted characteristics.

After choosing local neighbors (direct request receivers), TR assigns a time thresholds (TTL) to its message and sends it to them. They, on their turn, either propose an offer or propagate it to their own local neighbors following the same procedure as long as they have time (t<TTL). In that case, these agents acting as subcontractors (RR) send the feedback offer to $T_x$ as well as available ratings (see next subsection) for this partner. At the next step, $T_x$ assigns a value V, an indication of relevance, to each received trusted path, calculated as follows: $V = (pl-0.25*hp)*C_{RR}, \sqrt{pl} <= 5$ or $V = (pl-0.5*hp)*C_{RR}, \sqrt{pl} >= 6$ (2), where pl stands for the length of the trusted path, hp stands for the number of network nodes while $C_{RR}$ is the credit score of the local neighbor (RR agent) that returned that path. $C_{RR}$ is based on RR agent's credits with a time stamp that fits in TR requested time period. Using this time period, TR has a clue about RR's latest behavior. The V value discards feedback, taking into account risk.

*Establishing trust.*

Despite the chosen mechanism, as soon as, potential partners are located, $T_x$ proceeds to the next stage, establishing a relationship with the most promising eCommerce partner ($T_{SMicro}$ agent). We propose a reputation based approach. In general, reputation allows agents to build the degree to which an agent has confidence in another agent. Hence, reputation (trust) models help parties to decide who to trust, encouraging trustworthy behavior [3]. The core element here is the ratings, the evaluation reports of each transaction. According to our approach, a rating ($r \in ER_x$) is the fact: *rating*(*id→rating's_id, truster→truster's_name, trustee→trustee's_name, time→*t $EV_n^1 \rightarrow value_1$, $EV_n^1 \rightarrow value_2$, $EV_n^1 \rightarrow value_3$, $EV_n^1 \rightarrow value_4$, $EV_n^1 \rightarrow value_5$ $EV_n^1 \rightarrow value_6$, *confidence→?conf*), where *confidence* is agent's certainty for that rating while we recommend six well-known evaluation criteria for $EV_n^x$ values, namely *response time, validity, completeness, correctness, cooperation, confidence*.

The $T_x$ agent combines ($r_{i2}$ to $r_{i4}$) its own ratings (ER) with those received by recommenders (LOCATOR variation) to discard potential partners and next to estimate

the reputation for the remaining in order to find the most well-reputed among them. These rules are defeasible and they all conclude conflicting positive literals for the same pair of agents (truster and trustee). That's why there is a superiority relationship between the rules.

$r_{i2}$: *eligible_partner*(*rating*→*?id*$_x$, *cat*→local, *truster*→*?self*, *trustee*→ *?x*) := *rating*(*id*→*?id*$_x$, *truster*→*?self*, *trustee*→ *?x*, *confidence*→*?conf*$_x$).

$r_{i3}$: *eligible_ partner* (*rating*→*?id*$_x$, *cat*→*longerTie*, *truster*→*?a*, *trustee*→ *?x*) := *confidence_threshold*(*?conf*), *rating*(*id*→*?id*$_x$, *truster*→*?a*, *trustee*→ *?x*, *confidence*→*?conf*$_x$), *creditValue(?Vvalue), ?conf*$_x$ >= *?conf.*

$r_{i4}$: *eligible_ partner* (*rating*→*?id*$_x$, *cat*→*longestTie*, *truster*→*?a*, *trustee*→ *?x*) := *credit_threshold(?v), confidence_threshold*(*?conf*), *rating*(*id*→*?id*$_x$, *truster*→*?a*, *trustee*→ *?x, confidence*→*?conf*$_x$), *creditValue(?Vvalue), ?conf*$_x$ >= *?conf, ?Vvalue*>=*?v*, where category (cat) local refers to previously known agent, longerTie indicates a less than (5) path length and longest ties a greater path length (>5) and $r_{i2} > r_{i3} > r_{i4}$.

Due to superiority relationship, rule $r_{i2}$ indicates that personal opinion is important if there are ratings from itself (*truster*→*?self*,), otherwise logerTies opinion ($r_{i3}$) will be taken into account if it's confident (rating with confidence greater than $T_x$'s *?conf* threshold (*confidence*→*?conf*$_x$), *?conf*$_x$ >=*?conf*) whereas longestTies opinion ($r_{i4}$) should be confident (*?conf*$_x$>= *?conf*) and highly credit valued (*creditValue(?Vvalue), ?Vvalue*>=*?v*) based on $T_x$'s threshold (*?v)* in order to be taken into account.

After this discarding process, the final reputation value ($R_x$, x≡entity) of a potential partner, at a specific time t, is based on the weighted (either personal $W_{self}$ or recommended $W_{RR}$) sum of the relevant reports (normalized ratings in order to cross out extremely positive or extremely negative values) and is calculated as below. In the case that agent $T_x$ had chosen the broadcasting method, it uses only the first part of the above formula with its personal ratings for the agents that replied to its request.

$$R(t) = \frac{w_{self}}{w_{self} + w_{RR}} * \frac{\sum_{\forall t_{start} < t_i < t_{end}} \frac{w_n \; x \; log\left(r^{ER^n}\right) * t_i}{\sum_{n=1}^{N} w_n}}{\sum_{\forall t_{start} < t_i < t_{end}} t_i} + \frac{w_{self}}{w_{self} + w_{RR}} * \frac{\sum_{\forall t_{start} < t_i < t_{end}} \frac{w_n \; x \; log\left(r^{ER^n}\right) * t_i}{\sum_{n=1}^{N} w_n}}{\sum_{\forall t_{start} < t_i < t_{end}} t_i} \quad (3)$$

*Exchange data/request.*

Following this process, the $T_x$ decides upon the preferred SMicro agent (highest R value) and sends to that agent an ACCEPT-PROPOSE message ($r_{i5}$) indicating that it is interest for this proposal (*about*→*?proposeID*) as long as there is a stored offer (*propose* fact) from that agent. The message includes a time threshold (*tth*→*?t*) for the interaction, greater than current time (*?t*>$t_{current}$) specifying time availability. The SMicro agent, on its behalf, checks the offered service/product availability (*timeValid*→*?tv, specs*→*?specs*$_j$) and replies either confirming or withdrawing its offer.

$r_{i5}$: *send_message*(*msg* →*accept_proposal*(*about*→*?proposeID, tth*→*?t*), *sender*→*?self, receiver*→*?SMicro*) := *propose*(*id*→*?id*$_x$, *time* → *?tt, sender*→ *?SMicro, receiver*→ *?self, offer*→ *?offer, timeValid*→*?tv, specs*→*?specs*$_j$), *?t*>$t_{current}$.

*Negotiating terms.*

This stage is optional. If the $T_x$ agent is interest in negotiating specifications (*specs*) such as delivery time, it will continue by sending a negotiation request message to its $T_{SMicro}$ partner, which could accept or deny to negotiate depending on its strategy. The agents may terminate the negotiation without agreement at any point. In case of negotiation agreement, the message exchange sequence involves the following steps (repeated as many times as necessary): *Step 1*: $T_x$ (agent *i*) sends part of its belief clauses ${}^{i}B_R$ to the $T_{SMicro}$ agent (agent $_j$). *Step 2*: $T_{SMicro}$ agent evaluates the clauses ${}^{j}B_I$ (${}^{j}B_I \equiv {}^{i}B_R$) using its own beliefs ${}^{j}B_{R.}$ *Step 3*: $T_{SMicro}$ agent replies either with its conclusions (part of its inferred clauses ${}^{j}B_R$) or accepts the $T_x$ demand (${}^{i}B_R$ clauses).

*Reaching Agreement.*

Finally, the agents proceed with closing the eCommerce agreement. The $T_{SMicro}$ prepares the eContract, including the terms (*terms*→*?terms*) and the time it will be valid (*timeValid*→*?tv*, which should be greater than the signing time *?tv*>*t*), and sends a signing request message ($r_j$) to the $T_x$ agent specifying the offer (*about*→*?proposeID*), the time (*time*→*?t*) and the terms (*terms*→?terms).

$r_j$: *send_message*(*msg* →*signing_request*(*about*→*?proposeID, time*→*?t, terms*→?terms), *sender*→*?self, receiver*→*?x*) := *eContract*(*id*→*?id*$_x$, *time* → *?t, provider*→ *? self, client*→ *?x, terms*→ *?terms, timeValid*→*?tv), ?tv*>*t.*

## 3 Integration and Evaluation of the Methodology

The proposed methodology is integrated to EMERALD [6], a framework for interoperating knowledge-based intelligent agents. It provides, among others, a number of reputation models and reasoning services, among which four that use defeasible reasoning. These services are wrapped by an agent interface, the Reasoner, allowing other IAs to contact them via ACL messages. In essence, a Reasoner can launch an associated reasoning engine, in order to perform inference and provide results. The procedure is straightforward: each Reasoner stands by for new requests (*REQUEST*) and when it receives a valid request, it launches the associated reasoning engine that processes the input data (i.e. rule base) and returns the results (*INFORM*). As far as it concerns the language integration of the methodology, we use RuleML (included in the specifications of EMERALD) for our rules, representing and exchanging agent policies and clauses. We use the RDF model for data and belief representation. This methodology allows each agent to exchange its argument base with any other agent, without the need to conform to the same kind of rule paradigm or logic. Instead, via EMERALD, IAs can utilize third-party reasoning services, that will infer knowledge from agent rule bases and verify the results.

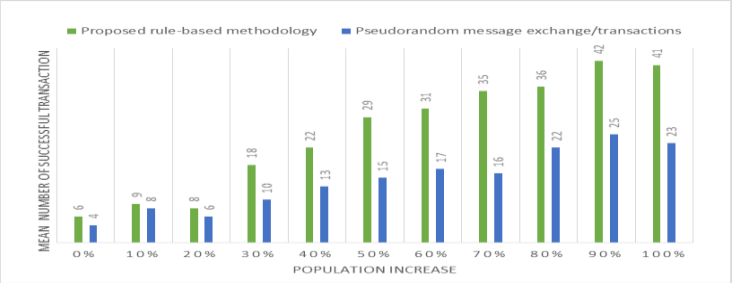


**Fig. 1.** Mean number of successful transactions.

As far as, it concerns evaluation we tried to simulate a realistic IoT environment, hence, we used 20% of $T_{hv}$ agents representing humans or virtual entities, 30% $T_{SMicro}$ agents representing web services and 50% of $T_{DMicro}$ agents that represent devices. The aim of the experiments was to calculate the mean number of successful transactions (ending to an eContract agreement) when using the proposed methodology, compared to a random transaction approach (Fig. 1). The adopted testbed consists of provider and consumer agents and it can be embedded to any eCommerce case while each experiment is populated with a different number of providers and consumers. From experiment to experiment we increased the number of agents about 10% in order to evaluate how the methodology behaves in various populated networks. We run eleven experiments; the first was populated with 20 providers and 20 consumers whereas the last was populated with 100 agents, divided in providers and consumers. The testbed includes good (15%), ordinary (30%), bad (40%) and intermittent providers (15%), namely honest and malicious agents. Good, ordinary and bad providers have a mean level of performance, hence, their activity (actual performance) follows a normal distribution around this mean. Intermittent agents, on the other hand, cover all possible outcomes randomly. As a result, ratings (and reputation values) vary in the network.

## 4 Related work

There are plenty approaches dealing with parts of the discussed topics, yet, there is still some lack to tightly related approaches combining a rule-based eCommerce perspective with microservices and agents for the IoT. In [8] a web service negotiation process is presented, focusing only on negotiation whereas our approach is more generic. The authors promote the reuse of the artefacts produced throughout the negotiation like our methodology which adopts the philosophy of clauses and policies reuse based on a rule-based mechanism. The authors support only web services opposed to our approach. As far as it concerns partner locating, Hang and Singh [2] also employ a graph-based approach but it focus only on measuring trust, with the aim to recommend a node in a social network using the trust network. The model uses the similarity between graphs to make recommendations. This approach similar to our LOCATOR variation attempts to take advantage of graphs in order to locate better partners, although this is just a part of our approach which takes into account more aspects in an attempt to sufficiently simulate eCommerce in the IoT.

## 5 Conclusion and Future Work

We presented a rule-based eCommerce methodology, forming a five-stage workflow, which allows Things to locate proper partners and establish trust relationships in the IoT network (there is no such support yet) via a social agent-based model for locating eCommerce partners and estimating their reputation. The study adopted agent technology, an increasing trend for IoT realization, combined the with the microservice architecture. A core concept of the methodology was the proper information exchange among heterogeneous Things, hence we proposed the use of semantic technologies such RuleML, although it is not yet adopted in IoT, in an attempt to support web evolution from Semantic Web to IoT and hopefully to the Internet of Agents. Finally, the methodology was integrated in EMERALD that provides appropriate Reasoners, supporting rule exchange with no common syntax. As for future directions, our intention is to enrich it with powerful mechanisms that will extract the relationships between potential partners as well as their past and future behavior. Hence, another direction is towards further improving it by adopting more technologies, such as ontologies, machine learning techniques and user identity recognition and management.

## Acknowledgment

The Postdoctoral Research was implemented through an IKY scholarship funded by the "Strengthening Post-Academic Researchers / Researchers" Act from the resources of the OP "Human Resources Development, Education and Lifelong Learning" priority axis 6,8,9 and co-funded by The European Social Fund - the ESF and the Greek government.

## References

1. Garriga, M.: Towards a Taxonomy of Microservices Architectures. Software Engineering and Formal Methods Lecture Notes in Computer Science, 203-218 (2018).
2. Hang, C., Singh, M.P.: Trust-based recommendation based on graph similarity. In AAMAS Workshop on Trust in Agent Societies (Trust), pp. 71-81 (2010).
3. Harwood, T., Garry, T.: Internet of Things: understanding trust in techno-service systems. Journal of Service Management, 28(3), 442-475 (2017).
4. Kontopoulos, E., Bassiliades, N., Antoniou, G.: Visualizing Semantic Web proofs of defeasible logic in the DR-DEVICE system. Knowl.-Based Syst., 24(3), pp. 406-419 (2011).
5. Kravari, K., Bassiliades, N.: Social principles in agent-based trust management for the Internet of Things. 14th Workshop on Agents for Complex Systems of 19th SYNASC (2017).
6. Kravari, K., Kontopoulos, E., Bassiliades, N.: EMERALD: A Multi-Agent System for Knowledge-based Reasoning Interoperability in the Semantic Web. 6th Hellenic Conf. on Artificial Intelligence (SETN 2010). LNCS, 6040/2010: 173-182 (2010).
7. Lee, I., Lee, K.: The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons 58(4), 431-440 (2015).
8. Silva, G.C., de Souza Gimenes, I.M., Fantinato, M., de Toledo, B.F.: Towards a Process for Negotiation of E-contracts Involving Web Services. SBSI 2012, pp. 267-278 (2012).