K^{4R}– Knowledge to the Power of RESTful, Resourceful and Reactive Rules

Ricardo Amador

CENTRIA, Universidade Nova de Lisboa, Portugal
http://centria.di.fct.unl.pt/

Abstract. The Web of today clearly answers questions of the form *"What is the representation of ...?"*. The Semantic Web (SW) of tomorrow aims at answering questions of the form *"What is the meaning of ...?"*. It is our stance that in order to realize the full potential of the original concept proposed by Tim Berners-Lee et al. (in Scientific American, May 2001), the SW must also answer, in a meaningful way, questions of a dynamic and active nature, like *"What to do if ...?"* or *"What to do when ...?"*. Moreover, SW questions of the form *"What to do ...?"* must be expressed and answered in a declarative, compositional and language agnostic way. It is our (hypo)thesis that formally established concepts, viz. the Web's REST architectural style, declarative SW representation of resources based on Description Logics (e.g., OWL-DL), and Reactive Rules (e.g., *"on* Event *if* Condition *do* Action*"* –ECA– rules), provide the proper theoretical foundations to achieve this goal. This paper describes our current research proposal, K^{4R} (pronounced, with an Italian flavor, *"Che fare?"*), towards achieving a declarative model for expressing (re)active behavior in and for the SW.

1 Introduction

Back in 2001, at the inception of the Semantic Web (SW), Tim Berners-Lee et al. [16] stated that the "real power of the Semantic Web will be realized when people create many programs that collect Web content from diverse sources, process the information and exchange the results with other programs". About a year later, James Hendler et al. [26] identified "the critical emerging application of the discovery and combination of web services" as a key SW technology. Despite this early recognition of the relevance of active components for the success of the SW, some years later, in 2006, the recap of the available SW technology [37] does not include a single mention to such active components. How is "[common] people [expected to] create many programs" and consequently realize the "real power of the Semantic Web"? This question remains unanswered and still relevant.

Though not mentioned in [37], much research effort has been dedicated in the last years towards enabling "the discovery and combination of web services" on a semantic level, e.g. [38]. The Semantic Web Services (SWS) Community has submitted several proposals to the W3C, viz. OWL-S, WSMO, SWSF and WSDL-S. To the present, the only W3C Recommendation concerning SWS is SAWSDL which contemplates only the semantic annotation of Web Services (WS). Such annotation makes functional building blocks (i.e. services) closer to the common user, but in the end current results

R. Meersman, P. Herrero, and T. Dillon (Eds.): OTM 2009 Workshops, LNCS 5872, pp. 789–799, 2009.

of the SWS approach do not relieve the common user of the burden of actually integrating those building blocks into tailored functional units of behavior. For this purpose, the SWS Community suggests automatic service composition though recognizing that "the Web service environment is highly complex and it is not feasible to generate everything in an automatic way" [36].

Let's take a "trivial example", from [16], which "occurs when Pete [a common user] answers his phone and the stereo sound is turned down". The building blocks for this behavior might be available as specialized services, viz. an event detection service that detects that Pete has answered his phone, a query service capable of collecting local devices with a volume control, and an active service that actually turns down the volume of such devices. Those services may be describable and discoverable on a semantic level, which makes Pete's task much easier. But how is Pete to say how he wants to use those services in order to achieve such a basic behavior? Pete is not a programmer, otherwise he would program a tiny service for that purpose. Pete does not know *how* to computationally achieve the behavior he wants, he only knows *what* is the desired behavior: "*when* a call is answered *if* there are devices to turn down *then* turn them down". Pete does not see this simple statement as an instance of a business workflow or as the result of some careful planning process. For Pete this statement is as simple as the rules he uses in his e-mail application to organize the messages he sends/receives.

Pete's intuition is correct. His statement has an obvious rewrite into a form of rules with a known declarative semantics: Event-Condition-Action (ECA) Rules. As such, Pete shouldn't need to program anything, all he should do is state the rule(s) of behavior that he wants his SW agent[1] to honor.

Providing the formal foundations that will allow Pete to do just that is the research challenge that we propose to address with the present research proposal. In the following, we start by clearly stating our research goal (Sect. 2), summing up other efforts related to it (Sect. 3), and proposing a concrete work plan to achieve our goal (Sect. 4). Section 5 further details how preliminary results lead us to the current proposal, and Sect. 6 discusses the originality and potential contribution of our proposal. Sect. 7 concludes the paper.

2 Objectives

Reactive Rules (RR) may be partitioned into Reactive Derivation Rules and Active Rules (cf. [2]). The former account for the definition of complex events/actions in terms of simpler ones and for establishing causal relationships from actions to events. The latter establish if and when to perform actions, and may take several forms, viz. Production Rules (PR) and several variations of ECA Rules. PR have the general form '*if* Condition *then* Action', and are commonly mistaken for Logical Derivation Rules, whose declarative semantics they do not share. The most common form of ECA Rules is '*on* Event *if* Condition *do* Action' but other forms, like $ECAP$ (which includes a post-condition to

[1] The term 'agent' is used in this paper in the same general sense as used in [16] or in [38], i.e. "as any software entity, possibly with problem-solving capabilities, including intelligent autonomous agents [27]".

be verified after the action) and EC^nA^n (with several guarded actions), have also been proposed.

The general *Research Goal* addressed by the present proposal:

- *includes* the definition of a computational model, based on RR, that will provide proper formal foundations to allow common people, lacking programming skills, to declaratively express (re)active behavior in and for the SW;
- *excludes* the (re)search for proper tools that are mandatory to make such a formal model actually usable by common people.

This general challenge, previously motivated in Sect. 1, is to be pursued under the following *Research Constraints*:

1. any expression of behavior in the SW *should* be transparent at the semantic level;
2. any expression of behavior in the SW *should* be made available, both at specification and evaluation time, in full conformance with the Web's REST architectural style [23] as instantiated by the Web's architecture and its HTTP[2] protocol;
3. expressions of behavior not obeying the previous constraints *must not* be excluded.

These constraints state that our (re)search *must* be for a computation model that is both Resourceful and RESTful, but *must not* withstand backward compatibility with current expressions of Web behavior (e.g. WS). Before proceeding, it is relevant to emphasize and clarify the actual implications of such constraints given our research goal:

- SW Reactive Rules should be fully transparent SW Resources with a mandatory abstract and formal Representation, i.e. the search is not for (yet) another XML markup for (reactive) rules, instead an OWL ontology is to be delivered;
- SW Reactive Rules should be Web Resources, i.e. they must be URI addressable (if persistent[3]) and accommodate distinct concrete Representations;
- full SW transparency must not be mandatory, semantical opacity must be allowed at all levels but always in a controlled way, e.g. a Reactive Rule component, like an event specification, must be allowed to be expressed in a SW opaque way —using some concrete markup or textual syntax— but its nature and interface must always be fully transparent;
- all the Resources included in our computational model must honor the REST uniform interface of the Web, i.e. the functionality of the computational model is to be instantiated into HTTP methods in full compliance with the HTTP specification;
- any extension to this uniform interface must be taken as a last resort to be avoided; if absolutely required, such extension, in order to be considered, must be properly identified, justified and rooted in proper theoretical foundations (e.g. [23,29]).

It is our stance that the previously stated goal is achievable under these constraints, i.e. HTTP, with a proper SW definition of Resources and the available standard Web representations, is more than enough to achieve our goal. SWS have their role to play when it comes down to actually specify, e.g., actions; however, a general model of Reactive Rules should be achievable without resorting to SWS.

[2] *HTTP - HyperText Transfer Protocol*, `http://w3.org/Protocols`.

[3] Unidentified resources are part of the SW, viz. RDF blank nodes. How to conciliate unidentified resources with the REST notion of (identified) resources is still an open challenge, as recent W3C notes show (e.g., *Cool URIs for the Semantic Web*, `http://w3.org/TR/cooluris`).

3 Related Work

The need to contemplate reactive behavior in the Web seems to be an undisputable matter. The clear recognition of such need dates, at least, back to 1998[4]. Throughout the years it has motivated several proposals on different levels and from different communities. The REST [23] community is aware of this open matter both on pragmatical[5] and theoretical levels (e.g. [29]). The WS community has even issued several standard proposals, cf. [40,39]. Proper standardization is still an open matter.

ECA Rules were intensively studied in the field of Active Databases (ADBMS), cf. [35], and are currently used in different fields like Workflow Management (WfMS), e.g. [30,11], and Multi-Agent Systems (MAS), e.g. [14,28]. In the MAS field [18] there are also attempts towards defining general models for evolution and reactivity including declarative approaches like [22,8].

ECA Rules were first proposed in the context of the SW back in 2003 [34], following previous XML-based proposals [12]. In the following years, between 2004 [33,21] and 2008 [9], considerable effort was dedicated, within WG-I5[6] of the REWERSE project, in order to achieve a model for evolution in the SW based on reactivity. This effort contemplated two distinct approaches: a language homogeneous approach instantiated into a concrete ECA rule language for the SW, viz. XChange[7] [20,19], and a language heterogeneous approach based on a general framework for ECA Rules in the SW [32,31]. The latter led to the development of two prototypes: MARS[8] [13,25] and r^3 [41,2]. MARS followed a markup-based approach where, e.g., languages are identified by XML namespaces, whereas r^3 followed an ontology-based approach where, e.g., languages and language symbols are OWL individuals.

Production Rules have also been proposed, as an alternative to ECA Rules [17], to model (re)active behavior in and for the Web. Standardization of different forms of Reactive Rules is currently under consideration by three standard bodies, viz. W3C[9], OMG[10] and RuleML[11].

4 Work Plan

In order to effectively pursue our research goal, previously stated in Sect. 2, the following *Research Hyphoteses* have been formulated and require confirmation:

1. RR are describable on an abstract/semantic level using currently standardized SW formalisms, i.e. RR can be true SW Resources;

[4] *WISEN Workshop on Internet Scale Event Notification, July 13-14, 1998, University of California, Irvine (UCI), USA*, `http://www.isr.uci.edu/events/twist/wisen98/`.

[5] *HTTP Subscription*, `http://rest.blueoxen.net/cgi-bin/wiki.pl?HttpSubscription`.

[6] REWERSE *WG I5 Evolution and Reactivity*, `http://rewerse.net/I5`.

[7] *XChange: A rule-based language for programming reactive behavior on the Web*, `http://rewerse.net/I5/XChange`.

[8] *MARS: Modular Active Rules for the Semantic Web*, `http://rewerse.net/I5/MARS`.

[9] *RIF Production Rule Dialect (RIF-PRD)*, `http://w3.org/TR/rif-prd`.

[10] *Production Rule Representation (PRR)*, `http://www.omg.org/spec/PRR`.

[11] *Reaction RuleML*, `http://ibis.in.tum.de/research/ReactionRuleML`.

2. such an abstract model of RR is an open model that embraces language heterogeneity, neither mandating nor excluding any specific concrete representation (e.g. some concrete markup);
3. such an abstract model fulfils the role of abstract syntax for the purpose of defining the semantics of RR;
4. RR provide a model of behavior for the SW with a declarative semantics;
5. such a declarative semantics is computationally groundable, or at least has a functionally equivalent computational model;
6. such a computational model/grounding is realizable in full conformance with the Web instantiation of the REST architectural style.

To attain confirmation of the previous hypotheses a *Research Work Plan* with *three phases* is proposed:

1. *Feasibility Assessment*. Achieve thorough understanding of RR on a computational level and in the context of the SW; additionally formulate a first proposal for an abstract description of RR using standardized Web formalisms.
2. *Declarative Definition*. Formal definition of a declarative semantics for RR based on an abstract syntax formalized using some form of Description Logic; evaluation of results with respect to the proposed hypotheses.
3. *Operational Implementation*. Computational grounding of declarative semantics; evaluation of results with respect to related work and results of the previous phase.

To realize the previous work plan different *Research Methods*[12] will be used in accordance with the nature of each phase:

- phases 1 and 3 are to be supported by the implementation of prototypes, and proper definitions based on Web standards (e.g. OWL ontologies and XML markups);
- phase 2 is to be supported by appropriate theoretical formalisms, viz. Model-theoretic semantics, Fixpoint Semantics, Kripke Structures and Description Logics;
- phase 3 may also require the support of theoretical formalisms for the definition of operational semantics, e.g. Transition Systems and Abstract State Machines;
- all phases are to be validated before proceeding to the next phase;
- phase 1 validation is to be realized against realistic application scenarios (e.g., [7]), by integrating different concrete languages (e.g., SPARQL, XQuery, WSDL2), as required by those scenarios;
- phase 2 shall be validated against existing rule standards (e.g., CL, JSR-94, PRR, RIF-BLD, RIF-PRD, RuleML) both for abstract syntax and semantics adequacy; phase 2 validation must identify any deviation from the research hypotheses, and any unavoidable deviation must be properly justified at this point;
- phase 3 is to be validated both on a pragmatical and theoretical level; on a pragmatical level, phase 3 prototype is to be validated for its Web RESTfulness and expressiveness against the scenarios of phase 1, and integration with other reactive platforms (e.g., EVOLP, MARS, Protune, ruleCore, XChange) is to be attempted; on a theoretical level, the computational grounding (as implemented by phase 3 prototype) is to be demonstrated to be conformant with the declarative semantics which resulted from phase 2.

[12] Together with [6] and the development infrastructure for the prototypes, the items mentioned between brackets constitute the most relevant *Research Material* required.

5 Current State

The present work started in 2005, when we first joined project REWERSE. At that point we had a WIDER[13] view of the challenge here presented. Such a WIDER view was targeted at the full research goal introduced in Sect. 2. It contemplated no exceptions, in fact our main focus was to be on tools themselves, not on the formal foundations. Throughout the duration of the REWERSE project we attained a better understanding of the actual complexity of our WIDER goal. After careful evaluation of the results of the project we came to the conclusion that we needed to focus on defining a formal declarative model, leaving WIDER perspectives open for the future. Most of the results currently available were obtained within the REWERSE project. Besides the author's key contribution towards defining the heterogeneous and ontology-based approach of REWERSE WG-I5, his main contribution to REWERSE was r^3 (i.e. Resourceful Reactive Rules).

r^3 is a prototype of a SW Rule Engine for ECA Rules which is capable of dealing with rules that use different languages either at the rule component level (event, condition, action), or within each component (by algebraic composition, based also on different algebraic languages). Such languages may range from general purpose to domain/application specific languages. At the heart of the r^3 prototype is the decision to fully embrace the SW and base its implementation on an RDF model. Every resource that matters to an r^3 engine is to be described using terms defined by a foundational OWL-DL ontology. Natively r^3 "talks" RDF/XML (using HTTP POST for communication, SOAP wrapped or not), but any other XML serialization of an RDF model is acceptable, provided an appropriate (bi-directional) translator is available (or implemented). Any request received by an r^3 engine is expected to be translated into an RDF model which is then added to an internal ontology that includes every resource known to a particular r^3 engine. This internal ontology must also be dynamically completed by means of automatic retrieval of missing pieces of information directly from the SW. Each r^3 engine directly supports a static set of languages. A Java library that abstracts, e.g., translation and communication details is available to help the integration of different languages as (static) components of an r^3 engine, either by implementing them from scratch or by wrapping existing implementations. More importantly, the set of languages supported by an r^3 engine is dynamically extendable live on the SW. Every r^3 engine is exposed online as an RDF resource. As soon as an r^3 engine becomes aware of another r^3 engine, the former fetches the RDF description of the latter which includes all the languages it supports (directly or indirectly). Consequently the former becomes a broker that indirectly supports also the languages supported by the latter. r^3 is available online (cf. [41]) including the integration of several languages, viz. EVOLP, HTTP, Protune, Prova, SPARQL, Xcerpt, XChange, XPath, XQuery, XSLT, and some utilities (e.g. send mail actions).

Since the first draft of our approach [5], all the main r^3 results were refereed and published [32,31,1,2]. A detailed account of those (and other more technical) results is included in [3] together with a detailed description of the realistic use-cases that were used to validate r^3 (e.g., [24]). The work described in [4], that led to the integration of

[13] *Web Integrated Development tools for Evolution and Reactivity (WIDER)*, http://code.google.com/p/wider3/wiki/WIDER.

Protune, introduces the general –and novel– concept of reactive SW policies and gave us the opportunity to further complement our validation of r^3 heterogeneity after the end of REWERSE.

Evaluation. Phase 1 of the proposed work plan is completed and validated. Following the research methods proposed in Sect. 1, all the objectives were accomplished, namely: a prototype of an ECA engine for the SW is available [41]; an OWL-DL ontology for RR is available [2]; and both the prototype and the ontology have been validated against realistic application scenarios (e.g., [24]) realized through the proper integration and use of several languages [3].

Most notably, and as far as we know, the results included in [2] constitute the only attempt to formally define and classify RR in an ontology. Nevertheless, the POST-based API used by r^3 does extend this ontology by adding concepts like *Message* (cf. [3]), thus introducing an undesirable level of complexity that comes close to a service-oriented approach; a clear resource-oriented (REST) approach will certainly reduce that complexity and avoid such r^3 specific concepts.

Work on phase 2 has started. The integration of r^3 with EVOLP and Protune raised some unanswered issues like representing explanations and answer-set models; the classification of RR included in [2] does not exclude (re)active forms of derivation rules which r^3 does not support; these issues when considered together under a clear REST constraint of a uniform interface (i.e. HTTP); led us to the conclusion that the declarative level envisaged for phase 2 required a more general definition of SW Knowledge Resources on an ontological level. For this purpose we have already defined the KR2RK ontology [10]. This ontology is yet to be integrated with [2] and properly validated against existing rule standards. That is the focus of our current work together with the definition of a declarative semantics that uses KR2RK as its abstract syntax.

6 Discussion

Reactive Rules provide a declarative and loosely-coupled model of behavior which, given the proper tools, may empower the common user to define tailored behaviors in and for the SW. This claim is supported by the success of PR in different fields of expertise, even if they express no declarative model of behavior. It is our stance that there is a semantical ambiguity in the condition component of PR that hinders their declarativeness: it is not clear if a Production Rule expresses an action to be taken when something *is true* or when something *becomes true*. From our point of view, this ambiguity is at the core of most semantical discrepancies among different PR implementations. The declarativeness of ECA Rules is much clearer and their current use by common users in specific fields, like mailbox management, is quite promising.

Since 2000 [15], the great majority of SW research efforts have been concentrated on querying the SW. Matters like evolution and (re)active behavior, though unanimously accepted as relevant in the context of the SW, seem to be forgotten. In this general state of affairs a few exceptions exist and are worth mentioning: the work of the SWS community, the work of WG-I5 of project REWERSE, and the efforts of the MAS community to demonstrate applicability of their results to the SW. Nevertheless when confronted with pragmatical choices most of these efforts seem to ignore that the SW is being built on strong theoretical concepts, namely REST and (logical) Knowledge Representation. When it comes down to make things work, foundational research work in the field of SW Evolution seems to be ruled by service-oriented and markup-based approaches, when it should be ruled by resource-oriented and ontology-based approaches. Oddly enough, we have come to a point where most standardization efforts concerning SWS are ontology-based, whereas all standardization efforts concerning rule languages for the Web, viz. RIF and RuleML, are mainly concerned with providing syntactical representations (i.e markups).

An ontology-based approach is particularly relevant if RR are to embrace language heterogeneity thus allowing, e.g., the use of domain/application specific languages that express concepts familiar to the user. Nevertheless, concrete representations of RR are required for this purpose. Among others, such concrete representations may be verbal, textual, graphical or markup-based; the SW supports all of them provided that eventually they are transparent on an abstract/semantical level. Such an abstract/semantical representation is in fact the only mandatory representation for SW Resources, and it is our stance that SW Rules are first of all SW Resources describable on logical terms like any other SW Resources.

Both r^3 and MARS demonstrate that such an heterogeneous and ontology-based approach is feasible. We are not aware of any other effort that comes close to modeling RR in such a language agnostic way. The most common approach to model RR, in the context of the SW, is to define a concrete rule language, thus limiting at the syntactical level the reactive expressiveness provided at the semantical level. A quick look at the specification of other proposals for (re)active rule languages for the SW, like RIF-PRD or XChange, is enough to identify a typical part of the syntactical specification where the supported events and actions are enumerated. In some cases, the provision for external actions and events is included, thus implying the use of some specific protocol. We are not aware of any proposal that relies on a RESTful approach for this purpose, i.e. none of the current proposals assumes that there is already a Web protocol (HTTP) which constitutes the uniform interface of the Web and that in order to use this protocol all that is required is the definition of resources and representations.

Modeling RR as true SW Resources includes acknowledging that they are also Web Resources that should honor HTTP has their uniform REST interface. We believe that no other protocol or interface, e.g. WS, is required to use RR as a general model of integration for SW behavior.

Contrasting with MARS (and with all other proposals) our proposal clearly makes two choices: REST and SW Resources. We are not concerned with services or with concrete resource representations. The declarative, heterogeneous, ontological and RESTful nature of our proposal, to model reactive behavior in and for the SW, make it a unique proposal in the current SW state-of-the-art.

7 Conclusion

The Web of today clearly answers questions of the form *“What is the representation of ...?”*. The Semantic Web (SW) of tomorrow aims at answering questions of the form *“What is the meaning of ...?”*. It is our stance that, in order to realize the full potential of the original SW concept proposed in [16], the SW must also answer, in a meaningful way, questions of the form *"What to do ...?"*. Such questions must be expressed and answered in a declarative, compositional and language agnostic way. It is our (hypo)thesis that formally established concepts, viz. the Web's REST architectural style, declarative SW representation of resources based on Description Logics (e.g., OWL-DL), and Reactive (e.g., ECA) Rules, provide the proper theoretical foundations to achieve this goal. Based on this hypothesis, in this paper we have presented our research proposal towards achieving a declarative model for expressing (re)active behavior in and for the SW.

Reactive Rules provide a loosely-coupled model of behavior and a component structure particularly suited to allow the semantically transparent expression of tailored behaviors through the composition of behavior fragments expressed using different languages. Given the proper tools and domain specific languages, that express concepts familiar to common users, such a model may even empower common users to define themselves the (rules of) behavior of their own personal (or professional) SW agents as envisaged in [16]. Since 2005, within REWERSE WG-I5, we have been "maturing" our language agnostic and ontology-based approach to Evolution and Reactivity in the SW, and a considerable amount of (preliminary) results has been achieved (cf. Sect. 5). These results have demonstrated that our approach is feasible.

More recently, after REWERSE, it became clear to us that the service-oriented complexity of these results was not mandatory. In fact, the most recent evolutions of our work (yet to be validated, cf. also Sect. 5), strongly suggest that by adding REST constraints to our approach, and consequently re-using the Web's uniform interface, we will be able to drop concepts of a more operational nature (e.g. *Message*, *Service* and *Interface*). To properly apply a REST approach more meaningful resources are required and those resources are the concepts that matter the most. For this purpose we have chosen to broaden the scope of our ontology-based approach, and extend it –on a general level– to other domains of Knowledge Representation (KR). The goal is not to model the specificities of every KR paradigm, but instead to establish a general foundational layer for KR supporting a resource-oriented approach, and then focus on the specificities of the paradigm of rule-based reactivity. The interested reader may find future evolutions of K^{4R} online at `http://k4r.googlecode.com/`.

Acknowledgements. This research has been funded, until Feb. 2008, by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779. Since March 2008, the author acknowledges the financial support of the Portuguese "Fundação para a Ciência e a Tecnologia" under "Programa Operacional Sociedade do Conhecimento". All the work presented and proposed here has been supervised by Prof. Dr. José Júlio Alferes, who acts as principal advisor for the doctoral studies of the author.

References

1. Alferes, J.J., Amador, R.: Towards a foundational ontology for reactive rules. In: ESWC 2007 (2007), `http://www.eswc2007.org/posters.cfm`
2. Alferes, J.J., Amador, R.: r^3 - a foundational ontology for reactive rules. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 933–952. Springer, Heidelberg (2007)

3. Alferes, J.J., Amador, R., Behrends, E., Franco, T., Fritzen, O., Krippahl, L., May, W., Schenk, F.: Prototype on the RDF/OWL level (2007), `http://rewerse.net/deliverables/m42/i5-d9.pdf`
4. Alferes, J.J., Amador, R., Kärger, P., Olmedilla, D.: Towards reactive semantic web policies: Advanced agent control for the semantic web. In: Bizer, C., Joshi, A. (eds.) ISWC 2008 (Posters & Demos). CEUR-WS.org, vol. 401 (2008)
5. Alferes, J.J., Amador, R., May, W.: A general language for evolution and reactivity in the semantic web. In: Fages, F., Soliman, S. (eds.) PPSWR 2005. LNCS, vol. 3703, pp. 101–115. Springer, Heidelberg (2005)
6. Alferes, J.J., Bailey, J., Berndtsson, M., Bry, F., Dietrich, J., Kozlenkov, A., May, W., Patranjan, P.-L., Pinto, A.M., Schroeder, M., Wagner, G.: State-of-the-art on evolution and reactivity (2004), `http://rewerse.net/deliverables/i5-d1.pdf`
7. Alferes, J.J., Berndtsson, M., Bry, F., Eckert, M., Henze, N., May, W., Patranjan, P.-L., Schroeder, M.: Use-cases on evolution, and reactivity (2005), `http://rewerse.net/deliverables/m12/i5-d2.pdf`
8. Alferes, J.J., Gabaldon, A., Leite, J.: Evolving logic programming based agents with temporal operators. In: IAT, pp. 238–244. IEEE, Los Alamitos (2008)
9. Alferes, J.J., May, W., Eckert, M.: Evolution and Reactivity in the Semantic Web. In: Semantic Techniques for the Web, The Rewerse Perspective. LNCS, vol. 5500. Springer, Heidelberg (2009)
10. Amador, R., Alferes, J.J.: Knowledge resources towards RESTful knowledge (March 2009), `http://k4r.googlecode.com/files/200903_kr2rk.pdf`
11. Bae, J., Bae, H., Kang, S.-H., Kim, Y.: Automatic control of workflow processes using eca rules. IEEE Trans. Knowl. Data Eng. 16(8), 1010–1023 (2004)
12. Bailey, J., Poulovassilis, A., Wood, P.T.: An event-condition-action language for XML. In: WWW 2002. ACM, New York (2002)
13. Behrends, E., Fritzen, O., May, W., Schubert, D.: An ECA engine for deploying heterogeneous component languages in the semantic web. In: Grust, T., Höpfner, H., Illarramendi, A., Jablonski, S., Mesiti, M., Müller, S., Patranjan, P.-L., Sattler, K.-U., Spiliopoulou, M., Wijsen, J. (eds.) EDBT 2006. LNCS, vol. 4254, pp. 887–898. Springer, Heidelberg (2006)
14. Berndtsson, M., Chakravarthy, S., Lings, B.: Result sharing among agents using reactive rules. In: Kandzia, P., Klusch, M. (eds.) CIA 1997. LNCS, vol. 1202, pp. 126–137. Springer, Heidelberg (1997)
15. Berners-Lee, T.: Opening Keynote: RDF and the Semantic Web. XML 2000, Washington, DC (December 2000)
16. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American, 29–37 (May 2001)
17. Berstel, B., Bonnard, P., Bry, F., Eckert, M., Patranjan, P.-L.: Reactive rules on the web. In: Antoniou, G., Aßmann, U., Baroglio, C., Decker, S., Henze, N., Patranjan, P.-L., Tolksdorf, R. (eds.) Reasoning Web. LNCS, vol. 4636, pp. 183–239. Springer, Heidelberg (2007)
18. Bordini, R.H., Braubach, L., Dastani, M., Fallah-Seghrouchni, A.E., Gómez-Sanz, J.J., Leite, J., O'Hare, G.M.P., Pokahr, A., Ricci, A.: A survey of programming languages and platforms for multi-agent systems. Informatica (Slovenia) 30(1), 33–44 (2006)
19. Bry, F., Eckert, M.: Rule-based composite event queries: The language xchange^{EQ} and its semantics. In: Marchiori, M., Pan, J.Z., Marie, C.d.S. (eds.) RR 2007. LNCS, vol. 4524, pp. 16–30. Springer, Heidelberg (2007)
20. Bry, F., Eckert, M., Patranjan, P.-L.: Reactivity on the web: Paradigms and applications of the language xchange. J. Web Eng. 5(1) (2006)
21. Bry, F., Furche, T., Patranjan, P.-L., Schaffert, S.: Data retrieval and evolution on the (semantic) web: A deductive approach. In: Ohlbach, H.J., Schaffert, S. (eds.) PPSWR 2004. LNCS, vol. 3208, pp. 34–49. Springer, Heidelberg (2004)

22. Costantini, S., Tocchio, A.: A logic programming language for multi-agent systems. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) JELIA 2002. LNCS (LNAI), vol. 2424, pp. 1–13. Springer, Heidelberg (2002)
23. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of California, Irvine (2000)
24. Franco, T., Alferes, J.J., Krippahl, L., Amador, R.: Bio-informatics reactivity features through the semantic web. In: Burger, A., Paschke, A., Romano, P., Splendiani, A. (eds.) SWAT4LS 2008. CEUR-WS.org, vol. 435 (2008)
25. Fritzen, O., May, W., Schenk, F.: Markup and component interoperability for active rules. In: Calvanese, D., Lausen, G. (eds.) RR 2008. LNCS, vol. 5341, pp. 197–204. Springer, Heidelberg (2008)
26. Hendler, J., Berners-Lee, T., Miller, E.: Integrating applications on the semantic web. Journal of the Institute of Electrical Engineers of Japan 122(10), 676–680 (2002)
27. Jennings, N.R., Sycara, K.P., Wooldridge, M.: A roadmap of agent research and development. Autonomous Agents and Multi-Agent Systems 1(1), 7–38 (1998)
28. Jiang, L., Liu, D.y.: A survey of multi-agent coordination. In: Arabnia, H.R. (ed.) ICAI 2006, pp. 65–71. CSREA Press (2006)
29. Khare, R.: Extending the Representational State Transfer (REST) Architectural Style for Decentralized Systems. PhD thesis, University of California, Irvine (2003)
30. Knolmayer, G., Endl, R., Pfahrer, M.: Modeling processes and workflows by business rules. In: van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, pp. 16–29. Springer, Heidelberg (2000)
31. May, W., Alferes, J.J., Amador, R.: Active rules in the semantic web: Dealing with language heterogeneity. In: Adi, A., Stoutenburg, S., Tabet, S. (eds.) RuleML 2005. LNCS, vol. 3791, pp. 30–44. Springer, Heidelberg (2005)
32. May, W., Alferes, J.J., Amador, R.: An ontology- and resources-based approach to evolution and reactivity in the semantic web. In: Meersman, R., Tari, Z. (eds.) OTM 2005. LNCS, vol. 3761, pp. 1553–1570. Springer, Heidelberg (2005)
33. May, W., Alferes, J.J., Bry, F.: Towards generic query, update, and event languages for the semantic web. In: Ohlbach, H.J., Schaffert, S. (eds.) PPSWR 2004. LNCS, vol. 3208, pp. 19–33. Springer, Heidelberg (2004)
34. Papamarkos, G., Poulovassilis, A., Wood, P.T.: Event-condition-action rule languages for the semantic web. In: Cruz, I.F., Kashyap, V., Decker, S., Eckstein, R. (eds.) SWDB 2003 (2003)
35. Paton, N.W. (ed.): Active Rules in Database Systems. Springer, New York (1999)
36. Rao, J., Su, X.: A survey of automated web service composition methods. In: Cardoso, J., Sheth, A.P. (eds.) SWSWPC 2004. LNCS, vol. 3387, pp. 43–54. Springer, Heidelberg (2005)
37. Shadbolt, N., Hall, W., Berners-Lee, T.: The semantic web revisited. IEEE Intelligent Systems Journal, pp. 96–101 (May/June 2006)
38. Sycara, K.P., Paolucci, M., Ankolekar, A., Srinivasan, N.: Automated discovery, interaction and composition of semantic web services. J. Web Sem. 1(1), 27–46 (2003)
39. Vinoski, S.: More web services notifications. IEEE Internet Computing 8(3), 90–93 (2004)
40. Vinoski, S.: Web services notifications. IEEE Internet Computing 8(2), 86–90 (2004)
41. Resourceful Reactive Rules (r^3), `http://rewerse.net/I5/r3`