Fuzzy Conformance Checking of Observed Behaviour with Expectations

Stefano Bragaglia[1], Federico Chesani[1], Paola Mello[1],
Marco Montali[2], and Davide Sottara[1]

[1] DEIS, University of Bologna,
Viale Risorgimento, 2 - 40136 Bologna, Italy
name.surname@unibo.it
[2] KRDB Research Centre, Free University of Bozen-Bolzano,
Piazza Domenicani, 3 - 39100 Bolzano, Italy
montali@inf.unibz.it

Abstract. In some different research fields a research issue has been to establish if the external, observed behaviour of an entity is conformant to some rules/specifications/expectations. Research areas like Multi Agent Systems, Business Process, and Legal/Normative systems, have proposed different characterizations of the same problem, named as the conformance problem. Most of the available systems, however, provide only simple yes/no answers to the conformance issue.

In this paper we introduce the idea of a gradual conformance, expressed in fuzzy terms. To this end, we present a system based on a fuzzy extension of Drools, and exploit it to perform conformance tests. In particular, we consider two aspects: the first related to fuzzy ontological aspects, and the second about fuzzy time-related aspects. Moreover, we discuss how to conjugate the fuzzy contributions from these aspects to get a single, fuzzy score representing a conformance degree.

Keywords: fuzzy conformance, production rule systems, expectations, time reasoning

1 Introduction

In the last ten years there has been a flourishing of models and technologies for developing, deploying, and maintaining ICT systems based on (heterogeneous, distributed) components. Paradigms such as Service Oriented Architectures, Cloud Computing, Business Process Workflows, have been exploited from the industry: nowadays, mature standards and solutions are available to the average customer, covering many of the ICT needs within the industry.

However, the complexity of such systems has grown pair-wise with the availability of such standards and tools. At the same time, the adoption of standards has fostered the use of heterogeneous (software/hardware) components. As a consequence, assuring the correct behaviour of such systems has become an important issue. To this end, approaches based on the notion of *conformance* have been proposed. Roughly speaking, the *expected* behaviour of the system is specified a-priori, by means of some formal language. Then, the complex system is observed at run-time, and the externally observed

R. Pirrone and F. Sorbello (Eds.): AI*IA 2011, LNAI 6934, pp. 80–91, 2011.

behaviour is confronted with the *expectations*. In case the expectations are not met by the observations, some alarms and/or managing procedures are triggered. With the term *conformance test* we refer to the process of evaluating if the observed behaviour matches the expectations, i.e., if the observed behaviour is *conformant* with the expectations.

Notably, the *conformance*-based approach has been object of intense research activity in many different application fields, especially when considering it in its most abstract way. In Multi-Agent Systems (MAS), for example, social approaches specify the agents' allowed interactions as expected behaviours (externally observed), and define violations in terms of deviance from what is expected. The framework SCIFF [1], as defined by Alberti and colleagues, is mainly focused on the notion of *expectations* and their violations. Commitments, as deeply investigated by authors such as Singh [21,9,25] or Colombetti and Fornara [13], concentrate on promises that arise as consequences of agent interactions: a debtor agent becomes committed towards a creditor (i.e., it is *expected*) to bring about (make true) a certain property.

In the Business Process field, for example, van der Aalst and colleagues have proposed declarative languages to focus on the properties that the system should exhibit: in the DecSerFlow language [18] the users can specify which are the business activities that are (not) *expected* to be executed, as consequence of previously (not) executed activities. Within the field of legal reasoning and normative systems, authors like Governatori and Rotolo [14] have proposed logic frameworks and languages to represent legal contracts between parties: the focus is on the *compliance* problem, and they evaluate it by establishing if the possible executions of a system are conformant with the legal aspects.

Most of the approaches investigated so far provide a boolean answer to the conformance problem. If the question is *"Is the observed behaviour conformant with what is expected"*, most of the systems generate only a simple answer of the type *yes/no*. However, taking inspiration from our everyday experiences, we argue that in many cases it is required a richer, more informative answer. Indeed, yes/no answers tend to oversimplify and to collapse the conformance check to only two possible values, while real situations would require an answer with some degree. In this sense, a score (a value comprised in the interval $[0, 1]$) would be a reasonable desiderata.

Let us consider, for example, an internet book seller who delivers items by mail. To reduce its costs, the seller often delegates the packaging and the shipping of the items to smaller book retailers, following a commercial agreement. The delivers must be conformant to some criteria established within the agreement. Then, the book seller performs a continuous monitoring of the delivery process. In particular, in this example we consider two aspects: $1)$ the quality of the packaging; and $2)$ the timing of the delivery. Both aspects contribute to establish if the delivering has been conformant with the expectations established in the business contract.

Evaluating the quality of the packaging means to take into consideration several different aspects, such as the use of a box of the right dimension, the water-sealing, the material used within the package, the care with which the items have been packaged, and many others[1]. Although a yes/no answer is still possible, it is reasonable to assign a score to the overall quality, thus capturing "how well" the packaging was done. Moreover, the evaluation criteria should be known a-priori (for example, they could be part of

[1] Ultimately, also the customer feedback would contribute to evaluate the packaging quality.

a business agreement). Such criteria would explicitly define the concept of "good package". Evaluating the quality of a particular packaging would consist of establishing if that package belongs, and with which degree, to the category of "good packages".

Similar observations could be done also for timing aspects. Again, answering with a yes/no answer to the question "was the package delivered in time?" is correct. However, in case of delays, there would be no way to evaluate how "big" was the delay. Depending on the business agreement (that sets the expectations), few or many days of delay would have a different impact when evaluating the conformance of that particular delivery (w.r.t. to the expectations).

All these examples suggest that a conformance test could be significantly enriched with evaluation scores. Given the *vague* and *gradual* definition of the desired constraints, it would be natural to exploit *fuzzy logic* for defining the notion of conformance. In this paper, we present a prototypical system, based on a fuzzy extension of the Drools rule-based framework [16], to evaluate in a fuzzy manner how much an observed event, fact or object, is conformant to a certain expectation. In particular, we focus on two different aspects: on one side, we tackle the problem of establishing if a certain event matches the expectation, by considering fuzzy ontological aspects on *what* happens and *what* is expected. On the other side, we concentrate on *temporal*-related aspects, and show how it is possible within Drools to define custom, fuzzy, time-related operators. Finally, we show also how it is possible to conjugate both the aspects (ontological and temporal ones) to get a single fuzzy evaluation. Such evaluation not only allows to provide a fuzzy answer to the conformance issue, but intrinsically supports also the ranking of the observed events/facts (w.r.t. expectations) on the base of the conformance criteria.

The actual idea behind this work is to show the feasibility of a hybrid rule-based and semantic approach to the conformance evaluation problem, with fuzziness added on top. We do not aim to define some new fuzzy theory, and we do not provide any new contribution to the fuzzy research field. Rather, we investigate how fuzzy logics can be used to characterize the conformance problem, and we do this by exploiting an existing (fuzzy) rule-based framework. While indeed trivial from a "fuzzy point of view", our approach is quite new, to the best of our knowledge, in other research fields like, e.g., Multi-Agent Systems.

2 Background on the Drools Framework and Its Support to Fuzzy Reasoning

Drools [16] is an open source system with the aim of becoming a "Knowledge modelling and integration platform". At its core, among several other tools, it includes a reactive production rule engine, which is is based on an object-oriented implementation of the RETE algorithm [12].

From a user perspective, the system offers a blackboard-like container, called *Working Memory* (WM), where the facts describing the "state of the world" can be `insert`-`ed`, `updated` or `retracted`. The rules, then, are *activated* accordingly whenever the WM is modified. A rule is an `IF-THEN` like construct, composed of a premise (Left Hand Side, LHS for short) and a consequence (Right Hand Side, RHS). The LHS part is composed by one or more *patterns*, which must be matched by one or more facts in the WM for the rule to become active. An active rule is then eligible to be fired, executing the actions defined in the RHS, which may either be logical actions on the WM or side effects. A pattern is a sequence of constraints a fact must satisfy in order to *match* with that pattern. Since facts are objects, in Drools, the first constraint is a *class* constraint, while the following ones are boolean expressions involving one or more object's *fields*.

2.1 Fuzzy Reasoning Capabilities

Drools' language is rich and expressive, but the core expressions that can be used in a LHS are equivalent to a boolean formula where the atomic constraints are linked using simple logic connectives such as `and`, `or`, `not` and the quantifiers `forall` and `exists`. The use of boolean logic accounts for high efficiency, but limits the expressiveness of the system.

In [23] an extension of Drools, namely Drools Chance, has been proposed to support fuzzy reasoning. The RETE engine has been extended to support (among other frameworks) fuzzy logics "in a narrow sense", i.e., to support special many-valued logics addressing vagueness, via truth degrees taken from an ordered scale. Under this assumption, the constraints are no longer evaluated to `true` or `false`, but generalized using an abstract and pluggable representation, called *degree*. Degrees can be concretely implemented using, among others, real numbers in the interval $[0, 1]$, intervals, fuzzy numbers, $\ldots$. Exploiting the fact that operators (*evaluators* in the Drools terminology) can be externally defined, the engine has been extended to allow the evaluators to return degrees in place of booleans. Moreover, logic connectives and quantifiers have also been made "pluggable" (externally defined) and configurable. Given many valued predicates, connectives and quantifiers, Drools Chance can evaluate the LHS of a rule in the context of a "narrow" fuzzy logic. Further extensions allow to support fuzzy logics in a "broader sense", and in particular to support linguistic variables and fuzzy sets in constraints, as introduced in [26].

2.2 Fuzzy Ontological Reasoning

In [5], a further extension has been built on top of Drools Chance, to support fuzzy semantic reasoning within the Drools rule system: in particular, the implementation of a fuzzy tableaux-based reasoner [3] is presented. This extension allows to define and reason on knowledge bases whose expressiveness is equivalent to the $\mathcal{ALC}$ fragment of the family of Description Logics.

The resulting framework (Drools, Drools Chance and the extensions for ontological reasoning) allows to import ontologies defined in a fuzzy manner,within the Drools WM. Moreover, it allows to write rules where the LHS pattern matching mechanism is extended with some (fuzzy) ontological evaluators. The framework allows to integrate, in a unified formalism and model, both ontological and rule-based reasoning, and both the aspects are treated in a fuzzy manner.

During the evaluation of the LHS of a rule, each (fuzzy ontological) statement in the LHS is associated with an *interval*, whose bounds are fully included into the range $[0, 1]$. The interval defines lower and upper bounds of the truth degree of the statement. Such

Listing 1.1. A rule using fuzzy and semantic statements

```
rule "Dispatching an order"
when
  $o : Order
  $c : Contact( this ~isA "retailer" )
       Score( retailer == $c,
              this ~seems "high" )
then
  /* Suggests $c for $o */
end
```

bounds can be respectively considered as the *necessity* (lower bound) and the *possibility* values (upper bound) of the truth degree for its related statement. The use of an interval (instead of a single value) has been chosen by the authors to overcome the dichotomy between Open World Assumption (OWA) and Closed World Assumption (CWA). Indeed, ontological reasoning is often performed assuming OWA semantics, while rule based systems such as Drools adopt the CWA semantics: such important difference makes impossible to define a unified semantics for both the reasoning paradigms.

The solution to the OWA/CWA issue proposed in [5] consists of assigning to each ontological statement an interval delimited by two fuzzy values, that are interpreted as the necessity and the possibility of the truth degree of the statement. In our interpretation, the necessity value (lower bound of the interval) corresponds to the OWA hypothesis, while the possibility value (upper bound of the interval) corresponds to the CWA hypothesis. To easy the use of the ontological statements and their truth value within a rule, two operators, POS and NEC, are provided to the user to extract OWA and / or CWA degree, respectively.

An example of rules integrating ontological statements and rule-based reasoning is shown in Listing 1.1. The meaning of such rule is the following: anytime she receives an order, the seller should choose a retailer to dispatch it. To this aim, assuming that her address book is an ontology where contacts are organized into (not necessarily disjoint) subclasses including “retailer” and that the system has a distinct component that returns a score on the service of each retailer according to customers’ feedback, the rule browses through the seller’s address book identifying all the contacts that qualify as retailers and whose score seems high. All the retailers so identified are suggested to the seller who can choose then the most appropriate retailer for the given order.

2.3 Exploiting Drools and Drools Chance to Manage Fuzzy Time Aspects

To implement the fuzzy reasoning capabilities (see Section 2.1), the authors of Drools Chance [23] have exploited a powerful feature of Drools: the possibility given to the users to provide their own definition of new evaluators. We have chosen to follow the same approach: on this line, we have defined new evaluators for treating temporal related aspects. Currently, we have not implemented any particular time-related formalism, although we are aware there are many in the literature. Probably the most famous logic about temporal aspects is the one proposed by Allen in [2], where a (crisp) logic for reasoning over temporal intervals, together with some operators, is provided.

Our current choice has been to implement ad-hoc, on-purpose time-related evaluators, to address the specific needs of our test-cases. Probably, for very simple cases, the same expressivity could be achieved directly using Drools fuzzy rules, or by means of a fuzzy ontology defining some basic time-related concepts. It is out of the scope of this paper to establish when some knowledge is better represented using a formalism rather than another one. Here we will stress only the fact that the notion of conformance has been always referred to *what* is expected towards *what* is observed. Recent works within the MAS research community, such as [24], have also stressed the importance of temporal aspects (such as, for example, deadlines). Our choice then has been to focus on ontological and time related aspects explicitly, thus making these two aspects "first class entities" in the notion of conformance.

3 Checking Fuzzy Conformance

Generally speaking, the conformance problem amounts to establish if the externally observed behaviour of a system/entity does respect (satisfy/fulfill) some given *expectations*. Thus, the notion of conformance is strictly related (and depends on) to the notion of expectation. Depending on the research field, conformance and expectations have been given different characteristics and flavours.

In our system we do not restrict to a particular notion of conformance and/or of expectation. We rather assume that the answer will be always a real number in the interval $[0, 1]$, and that such a number can be interpreted as *the extent to which* the observed behaviour is conformant to the expectation. Such generalisation allows to consider also previous "yes/no" approaches as particular cases.

Moreover, we consider the conformance test as the result of combining many, different aspects, each one contributing with its own fuzzy result to the final conformance degree. The number of aspects to be taken into consideration, and the methods/algorithms to evaluate each aspect, depend on the application domain and its modelling. In this paper we present some examples of conformance based on two different aspects: more precisely, the ontological aspect and the temporal one. Many other different aspects could be considered, such as, for example, geographical information.

Our notion of fuzzy conformance then is a two-level process, where at the lower level many single components (the evaluators) provide a fuzzy conformance degree related to each single aspect, while at the higher level each fuzzy contribution is combined to achieve a single fuzzy degree representing the overall conformance. To support such notion of conformance, then, many user-defined evaluators are needed. Single aspects need ad-hoc, specific evaluators properly designed for a particular aspect of the domain. For the evaluation of fuzzy ontological aspects, we resorted to use an existing extension of Drools Chance (see Section 2.2). For the time-related aspects instead, we provided our own implementation of the needed evaluators.

In the remain of this section we will discuss the example introduced in Section 1, and will show a possible implementation of our notion of fuzzy conformance. Briefly recapping the example, in the context of a business agreement, some local book stores perform the packaging and the delivering of items on behalf of an internet book seller. The seller continuously checks if the delivering of the packages is conformant with the commercial agreement. Indeed, such contract establishes which are the *expectations* about the items delivery. The local stores provides in a log file the description of the packaging, and the time it was delivered.

3.1 Conformance as Ontological Fuzzy Evaluation

Exploiting the extensions introduced in Section 2.2, our system supports the definition of fuzzy ontologies[2], i.e. of ontologies where, for example, individuals are instances of certain classes with a fuzzy degree. We can easily imagine then an ontology where the concept of GoodPackaging is defined as those individuals of the domain that are Packages (intended as atomic concept), and that have been water sealed, filled with bubble-wrap paper and carefully prepared. Of course, there would be cases of packages sealed without water-proof scotch tape, or packages where the items inside have not been rolled up with bubble-wrap. Such packages would belong with a low degree to the category of GoodPackaging. On the contrary, packages responding to all the requisites would be classified with the highest score to belonging to the GoodPackaging class.

From a practical view point, we aim to write a rule where in the LHS there is a pattern that evaluates how much a particular packaging was well-done. I.e., we want to add in the LHS a (fuzzy) ontological statement about a particular package being instance of the class GoodPackaging. The evaluation of the LHS would then compute the truth value of such statement, and such truth value could be then used within the rule itself (in the LHS as well as in the RHS). However, to exploit the support of fuzzy ontologies presented in [5], we have to deal with the fact that an interval bounded by two fuzzy values is given as answer, when evaluating fuzzy ontological statements. Such values represent the truth degree of the statement under the OWA and the CWA hypotheses: consequently, we must resort, on a domain-basis, to the OWA semantics (lower bound), or to the CWA semantics (upper bound).

Listing 1.2. A rule with a fuzzy ontological statement

```
rule "Fuzzy Ontological Matching"
when
    $p: Package( $p nec ~isA "GoodPackaging" )
then
    println($p.id + " isA GoodPackaging: " +
      Drools.degree);
end
```

Within the context of our example, we have depicted a quite simple fuzzy ontology: in this particular case using OWA rather than CWA would make no difference, since under both the semantics the statement would be evaluated with the same, exact score. In more complex situations however, where more complex representations of the domain would be taken into consideration, the choice of which semantics should be adopted is not a

[2] See [4] for an introduction to Fuzzy Description Logics, their representation, and many examples of fuzzy ontologies.

trivial task, and would largely depend on the particular application domain. Generally speaking, since we are applying such ontological reasoning to evaluate conformance, Open World Assumption semantics seems to be a safer choice, since it would support a "lazier" evaluation of the conformance. CWA semantics would support a more stricter notion of conformance, but with the risk of some "false positive" results.

In Listing 1.2 an example of a rule with an ontological statement in its LHS is presented. The LHS is evaluated every time an object representing a packaging in inserted in the working memory. The Drools engine then evaluates the ontological statement, where `~isA` is a shortcut for the classical instanceOf ontological operator; the "∼" symbols indicates that the statement is indeed evaluated within the fuzzy domain. The rules print out on the console the the truth value of the statement "`p instanceOf GoodPackaging`". Note also the use of the "`nec`" operator to select the lower bound (necessity) of the fuzzy ontological evaluation, corresponding to the OWA semantics.

3.2 Fuzzy Temporal Evaluation

We have already discussed in Section 2.3 how it is possible to easily extend DROOLS with new operators. Exploiting such possibility, it is possible to create fuzzy time-related evaluators on the basis of the needs for representing the domain. Let us consider again the example of the internet book store. In a crisp evaluation setting, the conformance would depend on the promised and the effective delivery date. If the latter follows the former, we could conclude that the deadline has not been respected, and the expedition is not conformant.

However, it makes sense to consider the expedition process with a larger perspective. One day of delay could be insignificant in certain situations, while could be a terrific problem in other situations. For example, if one-business day delivery is expected, one day of delay has a huge impact on the notion of conformance. Differently, if the delivery is expected within thirty days, one or two more days would have a more little impact on the evaluation of concept.

To support our example, we have defined a new simple evaluator that provides a fuzzy evaluation score of how much a deadline has been met or not. We named such evaluator `~InTime`, and it takes two parameters: $a)$ the difference t_d between the expected delivery date and the effective delivery date (zero if the delivery met the deadline); and $b)$ the duration of the interval of time t_e expected for the delivery (such expectation established, for example, when a customer finalized the order to the internet book seller). A possible definition of our evaluator could be the following:

$$InTime(t_d, t_e) = \left\lfloor 1 - \frac{t_d}{t_e} \right\rfloor_0$$

An example of a rule considering time-related aspects is presented in Listing 1.3.

3.3 Combining Different Fuzzy Contributions

Once the different aspects have been evaluated, the problem of deciding how to combine such different fuzzy contributions arises. Typically, such choice would be highly

Listing 1.3. A Drools rule evaluating in a fuzzy manner how much a temporal dealine has been respected

```
rule "Fuzzy evaluation of the delivery delay "
when
  Order ($e: expectedInDays)
  DeliveryLog( $d: delay ~InTime $e)
then
  println("Delivery is conformant with the" +
          " temporal deadline with score: " +
          Drools.degree);
end
```

dependent by the modelled domain. Again, Drools Chance offers the possibility of user--defining logical operators. Moreover, the most common fuzzy logic operators are natively supported by the framework, hence providing the user a vast range of possibilities. When defining new user fuzzy operators, a particular attention must be paid to define them in terms of simpler, available operators that ensure the truth functionality property: the aim is to guarantee that the resulting system has still an underlying semantics based on a (infinitely many-valued) fuzzy logic. Of course, if the chosen operators do not guarantee the truth-functionality property, then also the resulting system will not exhibit such feature.

Listing 1.4. A rule evaluating conformance in a fuzzy manner

```
rule "Fuzzy evaluation of conformance"
when
  Order ($e: expectedInDays )
  DeliveryLog(
    $d: delay ~InTime $e
    , @imperfect(kind=="userOp")
    $p: packaging nec ~isA "GoodPackaging")
then
  println("Degree of Delivery Conformance: " +
          Drools.degree);
end
```

In Listing 1.4 an example of a rule evaluating the conformance in a fuzzy manner is presented. The rule exploits the operators defined previously, and show how it is possible to evaluate the conformance in a fuzzy manner by means of Drools Chance and its extensions. In particular, we have considered here only aspects related to *what* is expected/observed, and *when* it is expected/observed. In a similar manner, the framework can easily extended to support many different aspects, adapting to the needs of the modelled domain.

4 Discussion and Conclusions

In this paper we have presented our idea of *fuzzy conformance*, motivated by the fact that the usual crisp notion of conformance might result too poor for capturing the vagueness and uncertain characteristics of real application domains. The use of (infinitely) many--valued fuzzy logic has been a natural choice for supporting such idea. Moreover, we propose to split the notion of conformance into the process of evaluating the conformance on many, different aspects, each providing its own fuzzy conformance degree, and then to combine such contributions into a single (fuzzy) conformance value. In particular, we focussed our attention on ontological and on time-related aspects. Nevertheless, we acknowledge that such a choice could result as too restrictive: for example, spatial/geographical aspects could be of a great interest when evaluating conformance in particular domains. Our approach is easily extendible towards such directions. From the practical viewpoint, we have shown how it is possible to easily exploit existing tools to implement such notion of conformance. In particular, we have used the Drools rule--based system, together with the Drools Chance extension, and a recent extension that supports fuzzy ontological reasoning. Then, we have discussed how, within such framework, it is possible to write rules that evaluates the conformance in a fuzzy manner.

The implementation of the presented framework is still in a prototypical stage, and further refinements will be addressed in the future. For example, we support conjunction of atomic expectations, combined with the temporal dimension: disjunctions of expectations are only partially supported, but we deem them as a fundamental feature. Also the expressivity of the supported ontology language will be addressed: our current implementation is based on [5], i.e. it supports $\mathcal{ALC}$ fragment, while we aim to exploit [4] for a greater expressivity. Moreover, a better assessment of the approach, using a real and complex scenario, together with an evaluation of the performances and a comparison with other solutions, is planned for the near future.

An important (open) issue is about using which one, and when, the right modelling formalism. For simple examples like the one used in the paper, it can happen that more than one formalism would be expressive enough for modelling that part of knowledge. E.g., the ontology described in Section 3.1 could have been represented also by means of custom-tailored rules, instead of a fuzzy ontology reasoner. Discussing which formalism is better for which situation is far behind the scope of this paper. However, we have adopted the following thumb rule: fuzzy ontologies have been used to describe the static structure and properties of the domain, while more dynamic aspects such as temporal (or spatial proximity) issues have been modelled by using/extending the rules.

Our work has many contacts with studies on the evaluation of constraints, temporal or not, under the generic notion of uncertainty. The concept can be applied at various levels of abstraction, from value-constrained measurements (e.g. [7]) to time--constrained interactions (e.g. [6], where the interactions are modelled by state transitions in automata). Many works on the evaluation of semantic constraints comes also from the Semantic Web area, especially to address the problems of service selection/ranking in general and of QoS evaluation in particular. In [20], the conformance is given a probabilistic connotation, while in [22] a fuzzy evaluation is performed. In [15], [17], finally, both semantic and fuzzy aspects about the QoS parameters are defined using an ontology, and evaluated using fuzzy predicates. Such solution, however, is based on a loosely coupling: examples of tighter levels of integration can be found for example in [19], where matchings are performed for recommendation instead of conformance check purposes.

With respect to the fuzzy research field, our work has many conjunctions with the well known idea of fuzzy pattern matching. In particular, our prototype share some similarities with the idea of weighted/tolerant fuzzy pattern matching [11,10], where a data item (composed of many features) is matched against an and/or formula (the pattern) of conditions. While in the fuzzy pattern matching both the pattern and the data can be vague/imprecise, in our scenario the data is certain and precise, since it results from the observation of the systems at run-time. Moreover, necessity and probability in our approach are mainly referred to the results of fuzzy ontological evaluation, and are a direct consequence of the ontology definition, while in the fuzzy pattern matching problem these terms refer to the vagueness introduced by the data (the observation in our case). Finally, researchers in the fuzzy area have focused on defining the operators so that the final resulting degree of matching will exhibit some properties, like for example preserving the semantics of necessity and possibility. In our prototype we leave the implementation of the operators to the domain modeller; we do not support directly the weights associated to each pattern's component, leaving the user the possibility of defining custom operators and, possibly, supporting multi-criteria evaluation policies.

We are aware that the presented approach juxtaposes with many other contiguous research fields, such as User Preferences specification or Recommender Systems. The possibility of having a degree of conformance allows to create a ranking of the observed behaviour, ordered on the basis of such conformance degree. If the evaluation of conformance is applied upon possible events/course of actions, instead of observed ones, we could get a sort of recommendation system. Investigating the relations with such existing research fields will be matter of future work.

Acknowledgements. This work has been partially support by the DEPICT Project, DEIS, and by the CIRI Life Sciences and Health Technologies, Univ. of Bologna.

References

1. Alberti, M., Chesani, F., Gavanelli, M., Lamma, E., Mello, P., Torroni, P.: Verifiable agent interaction in abductive logic programming: The SCIFF framework. ACM Trans. Comput. Logic 9(4), 1–43 (2008)
2. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM 26(11), 832–843 (1983)
3. Bobillo, F., Straccia, U.: fuzzydl: An expressive fuzzy description logic reasoner. In: IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2008 (IEEE World Congress on Computational Intelligence), pp. 923–930 (2008)
4. Bobillo, F., Straccia, U.: Fuzzy ontology representation using owl 2. International Journal of Approximate Reasoning (to appear, 2011)
5. Bragaglia, S., Chesani, F., Mello, P., Sottara, D.: A rule-based implementation of fuzzy tableau reasoning. In: Dean, et al. (eds.) [8], pp. 35–49
6. Crespo, F., de la Encina, A., Llana, L.: Fuzzy-timed automata. In: Hatcliff, J., Zucca, E. (eds.) FMOODS 2010. LNCS, vol. 6117, pp. 140–154. Springer, Heidelberg (2010), doi:10.1007/978-3-642-13464-712

7. De Capua, C., De Falco, S., Liccardo, A., Morello, R.: A technique based on uncertainty analysis to qualify the design of measurement systems. In: Proceedings of the 2005 IEEE International Workshop on Advanced Methods for Uncertainty Estimation in Measurement, pp. 97–102 (May 2005)
8. Dean, M., Hall, J., Rotolo, A., Tabet, S. (eds.): RuleML 2010. LNCS, vol. 6403. Springer, Heidelberg (2010)
9. Desai, N., Chopra, A.K., Singh, M.P.: Representing and reasoning about commitments in business processes. In: AAAI, pp. 1328–1333. AAAI Press, Menlo Park (2007)
10. Dubois, D., Prade, H.: Tolerant fuzzy pattern matching: An introduction. In: Bosc, P., Kacprzyk, J. (eds.) Fuzziness in Database Management Systems, pp. 42–58. Physica-Verlag, Heidelberg (1995)
11. Dubois, D., Prade, H., Testemale, C.: Weighted fuzzy pattern matching. Fuzzy Sets Syst. 28, 313–331 (1988)
12. Forgy, C.: Rete: A fast algorithm for the many patterns/many objects match problem. Artif. Intell. 19(1), 17–37 (1982)
13. Fornara, N., Colombetti, M.: A commitment-based approach to agent communication. Applied Artificial Intelligence 18(9-10), 853–866 (2004)
14. Governatori, G., Rotolo, A.: Norm compliance in business process modeling. In: Dean, et al. (eds.) [8], pp. 194–209
15. HongKang, Z., XueLi, Y., GuangPing, Z., Kun, H.: Research on services matching and ranking based on fuzzy qos ontology. In: 2010 International Conference on Computational Aspects of Social Networks (CASoN), pp. 579–582 (September 2010)
16. JBossL. JBoss Drools 5.0 - Business Logic Integration Platform (2010)
17. Lee, C.-H.L., Liu, A., Hung, J.-S.: Service quality evaluation by personal ontology. J. Inf. Sci. Eng. 25(5), 1305–1319 (2009)
18. Montali, M., Pesic, M., Aalst, W.M.P.v.d., Chesani, F., Mello, P., Storari, S.: Declarative specification and verification of service choreographiess. TWEB 4(1) (2010)
19. Ragone, A., Straccia, U., Noia, T.D., Sciascio, E.D., Donini, F.M.: Fuzzy matchmaking in e-marketplaces of peer entities using datalog. Fuzzy Sets Syst. 160, 251–268 (2009)
20. Rosario, S., Benveniste, A., Haar, S., Jard, C.: Probabilistic qos and soft contracts for transaction-based web services orchestrations. IEEE Trans. Serv. Comput. 1, 187–200 (2008)
21. Singh, M.P., Chopra, A.K., Desai, N.: Commitment-based service-oriented architecture. IEEE Computer 42(11), 72–79 (2009)
22. Sora, I., Lazar, G., Lung, S.: Mapping a fuzzy logic approach for qos-aware service selection on current web service standards. In: 2010 International Joint Conference on Computational Cybernetics and Technical Informatics (ICCC-CONTI), pp. 553–558 (May 2010)
23. Sottara, D., Mello, P., Proctor, M.: A configurable rete-oo engine for reasoning with different types of imperfect information. IEEE Trans. Knowl. Data Eng. 22(11), 1535–1548 (2010)
24. Torroni, P., Chesani, F., Mello, P., Montali, M.: Social commitments in time: Satisfied or compensated. In: Baldoni, M., Bentahar, J., van Riemsdijk, M.B., Lloyd, J. (eds.) DALT 2009. LNCS, vol. 5948, pp. 228–243. Springer, Heidelberg (2010)
25. Torroni, P., Chesani, F., Yolum, P., Gavanelli, M., Singh, M.P., Lamma, E., Alberti, M., Mello, P.: Modelling Interactions via Commitments and Expectations. In: Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models, pp. 263–284. IGI Global (2009)
26. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)