A Rule Engine for Relevance Assessment in a Contextualized Information Delivery System

Beibei Hu
Delft University of Technology
beibei.hu@tudelft.nl

Jan Hidders
Delft University of Technology
a.j.h.hidders@tudelft.nl

Philipp Cimiano
Semantic Computing Group, CITEC
Bielefeld University
cimiano@cit-ec.uni-bielefeld.de

ABSTRACT
In order to support police officers in their daily activities, we have designed a rule-based system which can deliver contextualized information to police officers, thus supporting decision making. In particular, we present a framework that has been designed on the basis of requirements elicited in a previous study, focusing on the rule language and engine that essentially defines and allows to configure the behaviour of the system. The rules consist of a body which specifies conditions that need to be fulfilled in a certain context. The head of the rules specifies how the relevance ratings of certain information items for specific users need to be updated given that the conditions in the body are met. On the basis of cumulated ratings, the system generates a user- and context-specific ranking of items. Quantitative evaluations in terms of precision and recall with respect to a gold standard determined in cooperation with police officers show that the system can cater for the requirements of our end users and yields reasonable precision and recall values.

Author Keywords
Context-aware information systems, rule-based systems, Semantic Web, RDF, SPARQL

ACM Classification Keywords
H.3.3 Information Storage and Retrieval: Query Formula- tion, Query Evaluation

INTRODUCTION
Evaluations of past incidents show that errors by public safety workers – police officers in particular – can be traced back to lack of critical information (see [1]). In one case, a policeman let a driver violating a red traffic light leave after merely warning him as he lacked information about the fact that the car had been stolen two days ago. This example suggests that police officers require context- and task-specific information to make quick decisions, but without being overwhelmed.

However, very few systems provide relevant and the appropriate amount of information meeting the specific information needs of police officers. In the context of the MOSAIC project (a Multi-Officer System of Agents for Informed Crisis Control1), we have developed a system which can assess the relevance of information items by taking into account contextual factors. The messages in our system can be regarded as small information containers consisting of facts – formalized in RDF2 – originating from different information resources. The processed messages are assumed to be shown to police officers on a mobile device, appropriately rendered for human consumption.

In this paper we describe a rule-based system which delivers contextualized information with the goal of supporting police officers in their daily activities. The major strength of the rule-based approach lies in its adaptability to new scenarios by mere reconfiguration of the rule set without the need of reprogramming. The main contributions of our research are (1) the design of a system which is able to assess the relevance of information according to the situation-specific requirements, (2) the definition of a simple rule language which is expressive enough to deal with the use cases we consider and allows to configure the behaviour of the system, and (3) the evaluation of the system using artificial scenarios to verify that the system can achieve reasonable precision and recall levels.

RELATED AND PREVIOUS WORK
There has been a lot of work on context-aware systems which tailor their behaviour to the needs of users acting in a specific context or environment [2, 6]. While using information about the location of users is a rather obvious choice, more and more approaches are moving beyond exploiting only location information and considering any information that can characterize the context, thus allowing systems to adapt their behaviour to specific situations (see for example the Service-oriented Middleware for building context-aware information systems described in [4] and the Contextualized User-Centric Multimedia Delivery System [7]). The system designed in the MultimediaN project for instance provides contextualized task allocation advice (see [8]), but does not tackle the problem of providing relevant information to users

1http://www.icis.decis.nl
2http://www.w3.org/RDF/
in a specific context. This is what we refer to as contextually informed delivery. Similarly to the agent-assisted email system [3], our system also provides a context-specific ranking of information items. Going one step further, our approach, by adopting a rule-based framework, can be easily configured and adapted to new situations.

In previous work, we have carried out a questionnaire-based user study in order to elucidate the information requirements of police officers based on constructed scenarios [5]. In this study, fifty fictive items were judged by police officers with respect to their relevance in specific phases of an artificially designed situation. The relevance levels considered were highly relevant, moderately relevant or irrelevant. The results of the study suggest that the users' information needs are not only task-dependent but also dependent on a specific phase of a situation. We build on the notion of a phase, understood as a sub-activity of a certain task that extends over a time interval. Such a phase could be investigation phase(5,12),(994,981) involving actions such as interviewing potential suspects. According to these findings a successful information system for assessing the relevant information for police officers has to take into account the task, the broader context but most importantly the specific phase of a situation to reliably determine the relevance of information items.

OVERALL SYSTEM
The system architecture and the information model is elaborated below.

System Architecture
The information items (also called “messages”) we consider are (i) real-time messages from emergency calls captured by the CCR (Command and Control Room), (ii) reports of police officers or (iii) background information contained in standard information systems such as criminal record databases. In addition, we use the term context data – captured via sensors or updated by the CCR who monitors task execution – is also assumed to be stored in our system. Context data provides information about the current situation of users, in particular about their task and activity in a given phase as well as about their current targets. An overview of how the system works is given in Figure 1. All formalized messages which contain the facts about events and background information are stored in the Message Repository. Once the context constructed in the Context Model is updated or a new message is received, the Rule Engine is triggered to execute all rules defined in the Rule Set to assess the relevance of messages in context. The ratings of all triggering rules are aggregated to yield cumulated ratings, which are stored in the Message Repository. The information items with the highest cumulated ratings are presented by the Message Presentation component according to some specified presentation strategy. This component is part of the overall system but is not discussed in this paper.

Information Modeling
We rely on the Resource Description Framework (RDF) as a data model in order to formalize the message and context information as it facilitates the integration of multiple data sources. The information model consists of two sub-models: the message model and the context model. The Message model represents the facts including the involved objects, targeted events, report time and location, and the type of information. The message model further captures the relevance of items for a given user via a Rating object with its properties isForUser and hasValue. The context model relies on an ontology-based context model to represent the users’ current situation by specifying the task at hand in a certain phase as well as the targeted events and objects. For instance, a police officer Joan, who is in phase of “ensuring preparedness”, is engaged in a task of “driving to the place” where a “car collision incident” is located. This incident, which happens at “10:00” and involves “drivers Bob and Jan”, is targeted by Joan. Here the italics indicate the properties in the model.

Rule Engine
This section describes the rule language and engine and how they are used to assess the relevance of information in a given situation.

Rule Example and Rule Syntax Specification
One rule that was derived from our previous user study can be expressed as: If one message is of type navigation information, and this message describes an event which is targeted by a user, and the user is engaged in the phase of ensuring preparedness, and the message is reported no later than 30 minutes before the task has started, then the message is highly relevant for the user. This example shows that the relevance of messages increases or decreases when a list of conditions is satisfied. Supposing that highly relevant corresponds to a numerical rating level of +5.0, the above example can be formalized in the rule language as follows:

(Rule body – Conditions)
(Rule head – Conclusion)
→ updateRating(+5.0)

Thus, the rules which specify the relevance of messages for users consist of a body and a head. The rule body contains a list of conditions that are combined together with the logical conjunction operator “&”, the union operator “|”, and
arithmetic as well as comparison operators for time variables (e.g. ?reportTime>?now - 20m, i.e. the message should have been received at most 20 minutes ago). The rule head determines how a relevance rating is updated. In the case that all conditions in the body are satisfied, the rating is incremented with the value indicated in the head (e.g. with the value +5.0 in our case). It is important to note that the values are simply summed up when the relevance for a given item is influenced by different rules concurrently.

Main Components
The rule engine has three components. (1) The rule parser parses a given set of rules and builds the abstract syntax tree (AST) according to the rule syntax. (2) The query generator recursively traverses the AST nodes in depth-first order and converts the whole tree to a SPARQL query. (3) The query evaluator evaluates the translated queries using the SESAME query engine and updates the aggregated ratings in the RDF repository. The relevance assessment process is shown in Figure 2. When the rule engine is triggered, it sequentially reads rules in the rule file. For each rule, the rule engine evaluates all conditions in the rule body. Given that all conditions are met, the relevance of a certain message is increased or decreased as indicated in the rule’s head. The cumulative ratings for a given user/message-pair are collected and updated accordingly after all rules are executed. A new ranking is then generated which is used to select an appropriate number of messages to deliver in the next step. The body of the above rule example is translated into the following SPARQL query. When evaluated, this query returns message/user pairs, i.e. (?message / ?user) bindings. The set of triple patterns in the WHERE clause represents the conditions that need to be fulfilled as specified in the body of the corresponding rule.

```
PREFIX cd: <http://ruleEngine.org/example/>
SELECT ?msg ?user
WHERE
{ ?msg cd:infoType cd:NaviInfo.
  ?event cd:targetedBy ?user.
  ?user cd:engagedIn ?task.
  FILTER (?reportTime > ?startTime - "1800000"^^xsd:long )
}
```

Figure 2. Relevance Assessment Process

<table>
<thead>
<tr>
<th>Time</th>
<th>Phase</th>
<th>Known Context</th>
<th>Task Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>22:00</td>
<td>Phase1: Ensuring Preparedness</td>
<td>Involved Drivers and Cars: Jan, driver of a car “SZ-VB-6”; Bob, driver of a car “01-GBB-1”.</td>
<td>Driving to the car collision spot within 10 minutes.</td>
</tr>
<tr>
<td>22:10</td>
<td>Rule</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22:25</td>
<td>Phase4: Tracing the Incident</td>
<td>Updated Context: Some suspect substances are found in the car “01-GBB-1”.</td>
<td>Inspecting the car “01-GBB-1”.</td>
</tr>
</tbody>
</table>

Table 1. The Car Collision Scenario Description

EVALUATION
Four scenarios corresponding to a different incident each were defined in cooperation with four police officers in a user study [5]. Two of them – a car collision incident and a fallen painter incident – are supposed to be the main events targeted by the participants. The other two “sideshow” scenarios are unrelated to the two main events and are included to construct a realistic situation including various events happening in parallel. In particular, some unrelated events could become relevant for a police unit if they happen close-by, thus potentially having some influence on the targeted event. All test persons spent one hour and fifteen minutes to directly judge the relevance of fifty information items. By reusing these four relatively realistic scenarios, the system has been evaluated with respect to a gold standard constructed in cooperation with police officers providing relevance judgements for a given phase.

Scenario Description
The scenarios are described as follows: A person called “Jan” is calling 112 explaining that his car with plate “SZ-VB-6” was hit by another car with plate “01-GBB-1” at the corner of the Mekelweg and the Steltjesweg in Delft around 20:00. Two minutes later a painter called “Freek” is calling 112 reporting that his colleague “Kees” has fallen down from the scaffolding at the Mekelweg 136. Two police units are dispatched, one unit per incident. During the course of the main events targeted by the police officers, two further sideshow events take place: a handbag robbery and a confused old man not being able to find his way. The history of the car collision scenario is sketched in Table 1.

Method
The system is evaluated w.r.t. the above described scenarios in the following steps. (1) Data construction: two datasets – $D_x$ and $D_y$ – are established, and each of them consists of 25 information items and contextual information for each of the main scenarios, i.e. handling the car collision and the fallen painter event. All the information is formalized by way of a total number of around RDF triples. For example, a message reporting the personal record of a driver Bob can be represented by such triple patterns: cd:msg1 cd:describesObject cd:driverBob. cd:msg1 cd:infoType cd:DrunkDrivingRec. (2) Rule definition: we spent around three hours to build an initial rule set $Rule_x$ consisting of thirty-two rules for the car collision scenario (about 5 minutes per rule). This shows that the effort for building rules is
reasonable. Moreover, we engineered the rule set $Rul_{x+y}$ by adding twenty new rules for the fallen painter scenario. (3)

**Evaluation:** we evaluate the performance of the system with respect to the top-5 and top-10 information items retrieved in terms of Precision/Recall and $F$-Measure in two evaluation modes: lenient and strict. In strict evaluation mode, we regard only items rated as highly relevant (i.e., with a rating of no less than +3.5) in the gold standard as correct (yielding the set $G_{\text{strict}}$), while in lenient evaluation mode we regard also moderately relevant items (rated with no less than +2.0) as correct (yielding the set $G_{\text{lenient}}$). Precision and Recall are defined as follows: $P_{\alpha}@k = |\text{Top} - k \cap G_{\alpha}|/k$, $R_{\alpha}@k = |\text{Top} - k \cap G_{\alpha}|/|G_{\alpha}|$ and $F_{\alpha}@k = 2P_{\alpha}@k \times R_{\alpha}@k/P_{\alpha}@k + R_{\alpha}@k$ ($\alpha \in \{\text{strict, lenient}\}, k \in \{5,10\}$).

As our system delivers information tailored to the particular phase, we average the above values for four phases per scenario. We evaluate the rule set $Rul_x$ on dataset $D_x$ and rule set $Rul_{x+y}$ on dataset $D_y$. In addition, we evaluate $Rul_x$ on $D_y$ to assess the performance of the old rules on a new scenario. To understand the interaction between the newly added and the old rules, we also evaluate $Rul_{x+y}$ on the dataset $D_x$.

**Results and Discussion**

The results of our system are shown in Table 2. By reconfiguring the behavior of the system based on two rule sets and two datasets, we can draw conclusions that (1) **the system is precise enough** to select relevant information as can be seen by testing the rule set on its target dataset. In fact, the precision ranges between 0.25 (top-10) and 0.7 (top-5) in lenient mode as well as between 0.73 and 0.9 in lenient mode, respectively. (2) **The coverage is adequate** when applying the rule set to its corresponding dataset, yielding a recall of between 0.9 (top-5) and 1.0 (top-10) in strict mode and between 0.5 and 0.92 in lenient mode, respectively. (3) **The rules developed for a specific use case are not overfitted**, as corroborated by the fact that the rule set $Rul_x$ provides a base performance corresponding to $F$-Measures of between 0.3 (F@10, strict) and 0.47 (F@5, strict) for scenario $D_y$. (4) **The addition of rules monotonically improves the system**, corroborated by the fact that the rule set $Rul_{x+y}$, a monotonic extension of rule set $Rul_x$ provides a comparable performance on the new scenario $D_y$ while not significantly degrading the performance on scenario $D_x$, compared to the rule set $Rul_x$. (5) **The effort for developing the rules is feasible** as the effort for developing the rule sets $Rul_x$ and $Rul_{x+y}$ amounts to 3 and 4.5 (3+1.5) hours, respectively.

**CONCLUSION**

In this paper, we have presented a rule-based system for the contextualized delivery of relevant information. In particular, a rule language has been defined which allows to specify the behavior of the system in a declarative fashion, thus allowing to modify the behavior of the system and adapting it to new scenarios without any programming effort. The rules are processed and executed by the rule engine and the ratings specified by all the rules triggered for a certain information item are cumulated to yield an aggregated score. Evaluations of the system with respect to a gold standard consisting of relevance assessments for fictive items provided by police officers shows that the precision and recall levels achieved by the system are very high. One clear limitation is that our system is only tested on limited data and lacks a mechanism that supports rule developers in adapting the system to new scenarios. In the future, we plan to evaluate the system on larger data in different domains and investigate the behavior of the system based on the rules written by police officers. Furthermore, we will expand the system by building the rules for the collaboration between colleagues and by devising a mechanism that facilitates the debugging and validation of the rules by domain experts.

**REFERENCES**


