First steps towards a context aware ontology-driven reporting system *

Mika Cohen
Swedish Defence Research Agency (FOI)
firstname.lastname@foi.se

Andreas Horndahl
Swedish Defence Research Agency (FOI)
firstname.lastname@foi.se

Christian Mårtenson
Swedish Defence Research Agency (FOI)
firstname.lastname@foi.se

ABSTRACT

In spite of constant technological advances, the nature of today's military conflicts has increased the importance of intelligence gathering by human observers. To allow for efficient exploitation, the information collected needs to be structured. Traditional hand-held reporting systems solves this by assuming a fixed report structure; the reporter submits a report by filling in the various data fields in a predefined report schema.

In this paper, we consider how to make the report schema aware of the reporting context and also dynamically adapt to information requests posed by actors external to the reporting situation: when a reporter enters data which are relevant to an information request, but do not fully answer the request, the schema expands to include fields that supply the missing information. We consider in detail how to evaluate relevance and how to select the additional data fields to query for.

When deciding what additional data fields to query for, we take into account not merely what data is missing, but also the information capability of the reporter, i.e., what kind of data the reporter is in a position to supply in the context he or she is in; more technically, we assume that the information capabilities of reporters are formalized in a simple epistemic logic extension to the domain ontology that underlies the reporting system, and use reasoning to infer the appropriate additional data fields for which to query the reporter.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5.0 [**Information Interfaces and Presentation**]: General; I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems

General Terms

Theory

Keywords

ontology driven user interface, context aware user interface, relevance matching, epistemic logic

1. INTRODUCTION

In spite of constant technological advances, the nature of today's military conflicts has increased the importance of intelligence gathering by human observers. To allow for efficient exploitation, the information collected from the human observers needs to be structured. A possible solution is to let the human observer input free text and then apply natural language processing techniques to transform the free text into the required structured format. However, such techniques are computationally intensive, often require a lot of training data and are never completely accurate. In a human reporting system these are limiting factors and alternative approaches are of interest.

In previous work [4], we introduced the concept of an ontology based adaptive reporting tool that supports a reporter when entering information about an observation. The reports are fed into a larger information system where they are further processed by software agents and consumed by other users. The information system uses an ontology as a common information model which the output of the reporting tool should adhere to.

The tool proposed in [4] lets the reporter enter structured information directly, but without assuming a fixed report structure; the tool can be seen as an adaptive form which reacts to user input. The tool is adaptive in the sense of adding new input fields and filtering out irrelevant fields based on the current user context (who is reporting, what is the role and capabilities of the reporter, where is the reporter, what time is it) as well as on possible information needs of other users in the information system.

Figure 1 provides an overview of how the reporting tool is intended to interact with external information needs. The reporter observes an event and enters some event information in the reporting system, which outputs semantic statements. These are then matched with information needs, *Requests For Information (RFIs)* from other parts of the system formulated as semantic queries. If there is a match, the reporter may be asked for additional information, i.e., asked to fill in additional input fields.

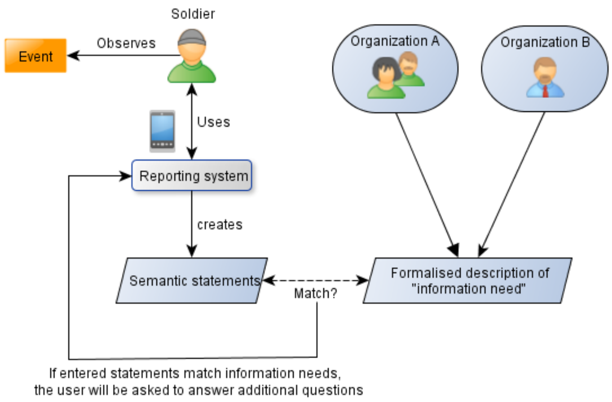

Figure 1: System overview

Asking for additional information might disturb the reporter and hence is associated with a cost. The problem of designing systems which tries to minimize this cost is studied in the field of intelligent user interfaces involving methods for so called polite interaction [1]. The goal there is to find the least disturbing moment to interrupt the user with a message or a question, which is a valid concern for our reporting tool.

However, in this paper we consider what additional questions - if any - should be presented to the user rather than at what specific point in time the user should be interrupted. To determine what to query the reporter we calculate the utility of posing a specific question based on the following criteria:

- The relevance of the question for satisfying one or more information needs;
- The ability of the reporter to supply the data asked for, given that the data exists;
- The probability that the data asked for exists;
- The value to the organisation of satisfying a specific information need.

The rest of the paper is organized as follows. In Section 2 we make a more formalized statement of our problem based on the criteria above. Section 3 reduces the problem of determining a query's relevance to an RFI (first item above) and the ability of a reporter to answer the query (second item above) to precise reasoning problems. In Section 4 we briefly discuss how to estimate the probability that the data asked for exists (third item above). Finally, Sections 5 and 6 concludes the paper with a general discussion and some thoughts on future work.

2. PROBLEM STATEMENT

In this paper we consider the problem of how to decide on-the-fly what additional input fields to present to a reporter in light of the input entered so far and in light of the RFI:s that analysts have posted to a central registry; we consider the question:

1. What additional input fields – if any – should we present to the reporter?

The answer to the question should balance the cost of disturbing the reporter with the expected benefit for analysts. Needless to say this balancing can be done in a number of ways, with different heuristics appropriate in different contexts.

EXAMPLE 2.1. *We introduce our running example. A new type of mine, MineTypeA, has recently appeared in a crisis region, RegionA. An analyst at HQ conjectures that a certain actor, actorA, is responsible. The analyst registers a request for information asking if MineTypeA detonates in close proximity to actorA. Eventually a reporter observes actorA and reports this. Soon thereafter and nearby, another reporter reports the observation of a mine. The reporting system considers the problem (1) and queries the reporter whether the observed mine was in fact a MineTypeA. The reporter confirms the query ('presses yes–button') whereupon the analyst at HQ is notified that MineTypeA has detonated in close proximity to actorA.*

Throughout the paper we assume an ontology (formal vocabulary) behind the reporting tool; text entered in input fields translate to formal assertions expressed with the vocabulary. In fact, we will ignore presentation issues and simply identify a report with a set of formal assertions.

In more detail, we assume a finite set of *classes* (unary predicates), a finite set of *properties* (binary relations), and a countable set of *individual constants*. An *assertion* is an expression of the form $a : C$ or of the form $a\,R\,b$, where a and b are individual constants, C is a class and R is a property. We assume furthermore a derivation relation $\vdash$ relating sets Δ of assertions to a single assertion F; we say that F is *derivable* from Δ if $\Delta \vdash F$. In practice, the derivation relation will be given by a finite representation – typically a proof system of some sort – that provides a procedure for computing the relation, i.e., deciding whether $\Delta \vdash F$ for arbitrary Δ and F.

For ease of presentation we do not assume any dedicated query language: a *query* about an assertion F is an expression $?F$; the query $?F$ is said to be true with respect to a knowledge base Δ if F is derivable from Δ.

The context for the decision problem (1) then is a report Δ_r just submitted by a reporter r, a central knowledge base Δ_{kb} (containing all reports submitted so far and, possibly, data from other sources, e.g., live database connections), and finally, a number of queries $?F_{rfi}$, $?F'_{rfi}$, … posted as requests for information by analysts.

EXAMPLE 2.2. *We revisit the above example with more formal details. We assume the domain ontology contains classes Mine, MineTypeA, atLocationA, etc. and properties closeTo, etc. satisfying e.g.,*

- *a:MineTypeA* $\vdash$ *a:Mine*
- *a: atLocationA, a': atLocationA* $\vdash$ *a closeTo a'*

for all individual constants a and a'*.*

To express his request for information, the analyst introduces a fresh assertion F_{rfi} *and extends the derivation relation such that:*

- *a:MineTypeA, a closeTo actorA* $\vdash$ F_{rfi}

(for all individual constants a) and registers the query $?F_{rfi}$ *as a request for information. Eventually, the first reporter submits the report: { actorA atLocationA }. Soon thereafter the second reporter, say* r_0*, submits the report: {* a_0*:Mine,* a_0*: atLocationA }, where* a_0 *is a fresh individual constant ('anonymous individual') generated for the report. The reporting system – after considering the question (1) – presents the reporter with the query:*

2. *?* a_0*:MineTypeA*

Once the reporter confirms the query, the request for information F_{rfi} *becomes derivable from the central knowledge base* Δ_{kb} *which stores all submitted data.*

In this paper we consider the decision problem (1) as the problem of computing on-the-fly a query $?\Delta$ (to pose to the reporter) with an expected utility $U(?\Delta)$ greater than a given threshold ϵ, i.e., $U(?\Delta) \geq \epsilon$; the rest of the paper is concerned with the details of the utility function U. At the most abstract level, we compute the expected utility of aksing the reporter $?\Delta$ as follows:

3. $U(?\Delta) = P(\Delta) \times 1_A(?\Delta) \times \sum_{?F} w(?F) \times 1_{R(?F)}(?\Delta)$

where $P(\Delta)$ is the probability of Δ being the case, 1_A is the characteristic function for the set A of queries $?\Delta$ which the given reporter is able to answer, $?F$ ranges over registered requests for information, $w(?F)$ is the weight assigned to the request for information $?F$ by analysts, and finally, $1_{R(?F)}$ is the characteristic function for the set $R(?F)$ of queries $?\Delta$ which are relevant to the information request $?F$.

According to the definition (3), the expected utility of querying the reporter $?\Delta$ depends, firstly, on the probability $P(\Delta)$ of Δ being the case - the intuition is that the query $?\Delta$ provides value only when the reporter replies affirmative. We do not consider negative evidence in this paper; a reporter can either confirm a query or reply 'unknown'.

Secondly, the expected utility of querying $?\Delta$ depends on the ability of the given reporter r to answer the query: the query provides no value if the reporter replies 'unknown', i.e., if $1_A(?\Delta) = 0$.

Thirdly, the expected utility of querying $?\Delta$ depends on the total weight assigned to all those RFI:s for which the query is relevant.

EXAMPLE 2.3. *We return to example 2.2 at the point when the reporter* r_0 *has just submitted her report. Should the reporting system add an additional input field querying the reporter whether* a_0 *is a MineTypeA? To decide this, the reporting system checks that the expected utility* $U(?a_0{:}MineTypeA)$ *of the query is greater than* ϵ*, the threshold. To calculate the expected utility, the system calculates:*

- $P(a_0{:}MineTypeA)$ *– the probability that* a_0 *is a MineTypeA*
- $1_A(?a_0{:}MineTypeA)$ *– if* a_0 *is a MineTypeA is the reporter able to confirm this?*
- $1_{R(?F)}(?a_0{:}MineTypeA)$ *– is the information that* a_0 *is a MineTypeA relevant for the request for information ?F?*

In subsequent sections we formalise the characteristic ability functions 1_A, the characteristic relevance function $1_{R(F)}$, and the probability function P.

3. COMPUTING RELEVANCE AND ABILITY

In this section we formalise the characteristic ability function 1_A and the characteristic relevance function $1_{R(?F)}$ from (3) as precise reasoning problems.

Informally, a query $?\Delta$ is relevant to a request for information $?F$ if an affirmative reply to the query resolves the request for information. Formally, we define $1_{R(?F)}(?\Delta) = 1$ if and only if F is derivable from Δ together with existing data, i.e., $\Delta, \Delta_{kb} \vdash F$; else $1_{R(?F)}(?\Delta) = 0$.

EXAMPLE 3.1. *Returning again to example 2.3, the query* $?a_0{:}MineTypeA$ *is relevant to the request for information* $?F_{rfi}$*, in symbols* $1_{R(?F_{rfi})}(?a_0{:}MineTypeA) = 1$*, if and only if* $a_0{:}MineTypeA$*,* $\Delta_{kb} \vdash F_{rfi}$*, i.e., if*

- a_0*:MineTypeA,* $\Delta_{kb} \vdash a_0$*:MineTypeA*
- a_0*:MineTypeA,* $\Delta_{kb} \vdash a_0$ *closeTo actorA*

The first condition is immediate. The second condition holds since the knowledge base Δ_{kb} *includes data from the earlier report, namely actorA: atLocationA, as well as data from the present report, namely* a_0*: atLocationA.*

Next we consider how to formalise also the characteristic ability function 1_A as a reasoning problem, first as a lightweight reasoning problem and then as a more heavy-weight reasoning problem.

3.1 Light-weight reasoning about ability

Informally, the given reporter is able to answer the query $?\Delta$ if, whenever Δ is the case the reporter is in a position to verify that Δ is the case. Formally, we define $1_A(?\Delta) = 1$, if and only if, Δ is a *specialisation* of the current report (entered data) Δ_r, i.e., Δ can be obtained from Δ_r by substituting more specific concepts/properties for less specific ones and, possibly, removing some assertions.

In detail, a concept C is more specific than a concept C if $a : C \vdash a : C$ for all individual constants a. Analogously, a property R is more specific than a property R' if $a\,R\,b \vdash a\,R'\,b$ for all individual constants a and b.

EXAMPLE 3.2. *Continuing example 2.3, the reporter is able to answer the query $?a_0$:MineTypeA, in symbols 1_A($?a_0$:MineTypeA) = 1, if a_0:MineTypeA is a specialization of the current report Δ_r, i.e., if MineTypeA is more specific than Mine.*

3.2 Epistemic reasoning about ability

The intuition in Section 3.1 is that the reporter is able to supply more specific classes and properties if asked. For example, it is assumed that if an instance of *MineTypeA* is known by the reporter to be a *Mine*, then it is also known to be a *MineTypeA*:

4. *a:MineTypeA*, *r knows a:Mine* $\Longrightarrow$ *r knows a:MineTypeA*

The assumption (4) is of course over overoptimistic and may cause the reporting system to disturb reporters unnecessarily with queries they are unable to answer. In particular, while the assumption (4) may be reasonable for reporters in mine squads it is perhaps unreasonable for reporters in relief units.

Next we formalise the characteristic ability function 1_A as a reasoning problem using assumptions that are custom−made for a specific context. The method requires manual preparation − creating the custom−made assumptions − but in return may offer less noise, i.e., reduce the number of queries that reporters are unable to answer.

Assumptions about reporters − assumptions about what this or that reporter knows in this or that situation − are made explicit in a context specific extension to the domain ontology (the ontology underlying the reporting system).The ontology extension is built using epistemic assertions of the form *r knows F*, where r ranges over a given finite set of reporter names and F is a regular assertion (assertion in the domain ontology language).[1] Intuitively, the epistemic assertion *r knows F* states that the reporter r is in a position to infer (verify) that F holds.

EXAMPLE 3.3. *Returning to example 2.2, someone − say an 'administrator' of the reporting system − extends the domain ontology with assumptions about reporters, such as the assumption that mine experts are able discriminate between different types of mines:*

5. *a:MineTypeA, r knows a:Mine, r:MineExpert*
   $\vdash$ *r knows a:MineTypeA*

However, the administrator does not extend the domain ontology with the stronger assumption (4).

Given the assumptions about reporters in the extended ontology, we can use reasoning to compute queries $?\Delta$ which a given reporter r is able to answer, i.e., queries $?\Delta$ such that if Δ is the case then r *knowst* Δ.

Formally, we define $1_A(?\Delta) = 1$, if and only if, $\Delta_{kb} \vdash \Delta \rightarrow$ *r knows* Δ, i.e., if and only if, $\Delta_{kb}, \Delta \vdash$ *r knows* Δ.

EXAMPLE 3.4. *Continuing example 2.3, the reporter is able to answer the query $?a_0$:MineTypeA, in symbols 1_A($?a_0$:MineTypeA) = 1, if a_0 :MineTypeA $\rightarrow$ r knows a_0 :MineTypeA is derivable from existing data, i.e., a_0 :MineTypA, $\Delta_{kb} \vdash$ r knows a_0 :MineTypA.*

When reasoning about the knowledge of reporters we follow standard practice in knowledge representation and assume that reporters are perfect reasoners and know all logical consequences of what they know (cf. [2]):

6. If $\Delta \vdash F$, then *r knows* $\Delta \vdash$ *r knows F*

(Note that the intended meaning of *r knows F* is that the reporter r is able to verify that F holds if asked.) In addition to (6), we assume the extended ontology inherits all proof rules from the domain ontology.

EXAMPLE 3.5. *As a special case of the assumption (6) about perfect reasoners, if class C is more specific than C' then r knows a:C $\vdash$ r knows a:C'.*

We assume that the knowledge base Δ_{kb} is extended with context data about reporters, such as an assertion stating that a certain reporter is a mine expert. In addition, we assume the extended knowledge base keeps track of who has said what: the knowledge base includes assertions *r knows* Δ_r for every report Δ_r submitted (now or in the past) by the reporter r.

EXAMPLE 3.6. *Continuing example 3.4, we assume the knowledge base Δ_{kb} contains the context data: r_0:MineExpert. Once the reporter r_0 submits her report { a_0:Mine, a_0: atLocationA }, the knowledge base Δ_{kb} contains: r_0 knows a_0:Mine, r_0 knows a_0: atLocationA.*

EXAMPLE 3.7. *Continuing examples 3.4 and 3.6, it follows from the custom asumption (5) that the reporter is able to answer the query, i.e., a_0 :MineTypA, $\Delta_{kb} \vdash$ r knows a_0 :MineTypA. However, if the knowledge base had not included the context data that the reporter r_0 is a MineExpert, then it would not follow that r_0 is able to answer the query.*

[1] We assume the set of reporter names is a subset of the set of individual constants.

4. COMPUTING PROBABILITY

In this section we consider how to obtain the probability $P(\Delta)$ when calculating the expected utility of a query $?\Delta$ according to (3).

Intuitively, the (subjective) probability of the event Δ is the conditional probability of Δ given everything the system knows, i.e., given the current knowledge base: $P(\Delta) = P(\Delta \mid \Delta_{kb})$. Unfortunately, estimating the conditional probability between arbitrary sets Δ and Δ_{kb} is notoriously difficult (even if the ontology is fixed).

Here, we consider a light-weight approach that places little demand on representative traning data or user input from domain experts. Since the characteristic ability function $1_A(?\Delta)$ in (3) is equal to 0 unless Δ specialises the current report Δ_r, we can assume that Δ is obtained from the current report Δ_r by replacing classes $C_1, C_2, \ldots$ with the more specific classes $C_1', C_2', \ldots$ and replacing properties $R_1, R_2, \ldots$ with the more specific properties $R_1', R_2', \ldots$.

We define the probability $P(\Delta)$ as the product:

7. $P(\Delta) = \prod_i P(C_i' \mid C_i) \times \prod_i P(R_i' \mid R_i)$

of all the conditional probabilities $P(C_i' \mid C_i)$ and $P(R_i' \mid R_i)$. Intuitively, the definition (7) approximates the conditinal probability $P(\Delta \mid \Delta_{kb})$ given the knowledge base with the conditional probability $P(\Delta \mid \Delta_r)$ given the current report, which in turn is computed under the simplifying assumption that assertions are statistically independent.

We consider three methods of obtaining the conditional probabilties $P(C_i' \mid C_i)$ and $P(R' \mid R)$ used in (7). The first method is to ask domain experts, which might be feasible if e.g. the number of classes is small and experts are available.

The second method is a straight-forward frequency analysis using available instance data, either from the knowledge base or from a training set:

8. $P(C' \mid C) = \frac{|\{a:C'\}|}{|\{a:C\}|}$

where $|\{a : C\}|$ is the number of instances of the class C in the available data set ($P(R' \mid R)$ is estimated analogously). Of course, the frequency analysis (8) requires a data set which is reasonably respresentative for the application domain.

The third method, on the other hand, requires neither representative data sets nor input from experts. The method simply assumes that instances are equally distributed among (immediate) subclasses to a common superclass:

9. $P(C' \mid C) = \frac{1}{|\{X<C\}|}$

where $|\{X < C\}|$ is the number of immediate subclasses to C according to the domain ontology and C' is itself an immediate subclass of C. The definition (9) is lifted to conditional probability for arbitrary subclasses C' (not merely direct sublasses of C) by taking the product of the conditional probabilities along the path from C to C' in the subclass-graph.[2]

The same product construction can be used also in the first- and second method above in order to reduce the number of conditional probabilities $P(C' \mid C)$ that need to be determined, reducing the burden on domain experts and representative training data.

5. DISCUSSION

Ideally, the reporting system has access to a rich description of each reporter's abilities and enough representative data to calculate the required probabilities. Unfortunately, this is not always the case, and approximations and heuristics such as those presented in the previous sections will be necessary.

Since reporting situations can vary quite dramatically, the proposed heuristics may result in unpredictable variations in the estimated utility levels, leading to large variations in the number of questions posed to the reporter. To avoid the risk of overload, a dynamic utility threshold based on how well the adaptable component is performing could be useful. Another solution could be to simply limit the number of questions to ask the reporter, regardless of their expected utilities.

So far we have not addressed the problem of where in the reporting process to present questions to the user. Research related to human-machine interaction has shown that interrupting users without taking their current situation and task in mind can lead to annoyance [1]. One aspect of particular interest to our application is how long to wait before presenting a query after it has passed the utility threshold. In the case where the user is already about to enter the information that the system wants to ask about, interruption is unneccessary and might be conceived as annoying. On the other hand, if the system waits too long before asking the question, the user might have switched focus and the interruption will be even more disturbing.

Instance subtype prediction based on the ontology structures is associated with assumptions about the type distribution. The simple and naive method where we assume that the subtypes are equally distributed may work in some cases but may give an incorrect, and perhaps misleading, representation of the probability distribution in other cases.

An alternative approach might be to replace probability in the utility function (3) with other less precise metrics that do not depend on assumptions about probability distributions. One such metric is semantic distance which (in its simplest form) counts the number of edges in the path from one concept to another more general concept in the ontology class hierarchy. Semantic distance per se does not say anything about how probable it is that an instance of one type actually is an instance of a more specific type. Nevertheless,

[2] It is assumed that the subclass-graph is a tree, as is often the case for tradtitional light-weight ontologies such as taxonomies, directories, etc.

one could perhaps assume, as a heuristic, that a large distance indicates a low probability. However, one would have to interpret the semantic distance in terms of probability in order to use it in the utility function (3). Interpreting semantic distance in terms of probability would force us, as all subjective probabilistic approaches, to make assumptions and use heuristics to model the necessary distributions. This might turn out to be very difficult.

The matching problem, the problem of determining if a report is a potential answer to an RFI, could alternatively be solved using an RDF graph matching approach. More specifically, the report (represented by an RDF graph) could be compared with the query (represented as an RDF graph with variables) using concept similarity measures like the ones described in [5]; if the report is similar to the query, the RFI itself is presented as a query to the reporter. While simple and straightforward, this approach does not take into account the fact that the reporter might not (by herself) be able to supply all the information asked for in the RFI. Thus, exposing the complete RFI to the reporter may lead to considerable noise, i.e., queries that disturb the reporter without benefiting the analyst.

As shown in the previous section, we can exploit the ontology structure to estimate the probability that an instance belongs to a specific class. The problem of predicting the type of a new instance can be seen as a standard classification problem where machine learning algorithms can be applied. However, selecting which features to use to train a classifier is not always straightforward. In the previous section we based the predictions solely on the super-class, but other features such as instance data can be used. The generic prediction problem in such a semantic setting can be formulated using statistical relational learning (SRL) as presented in [3].

6. CONCLUSIONS AND FUTURE WORK

In this paper we presented the concept of an ontology-based adaptive reporting tool that works as a broker between reporters and analysts with information needs. We analysed in detail how the reporting tool should decide on-the-fly whether to ask a reporter for some additional information, given the context of a reporting situation and a set of external information needs. The proposed formal solution balances the expected utility of receiving a relevant answer with the cost of disturbing the reporter. The described approach was only explored theoretically. The next step is to do an implementation and perform user tests as described in [4].

Acknowledgement This work was supported by the FOI research project 'Tools for information management and analysis', which is funded by the R&D program of the Swedish Armed Forces.

7. REFERENCES

[1] G. S. Bahr and R. A. Ford. How and why pop-ups don't work: Pop-up prompted eye movements, user affect and decision making. *Computers in Human Behavior*, 27(2):776 – 783, 2011. Web 2.0 in Travel and Tourism: Empowering and Changing the Role of Travelers.

[2] Y. M. Joseph Y. Halpern, Ronald Fagin and M. Y. Vardi. *Reasoning about knowledge.* MIT Press, 1995.

[3] C. Kiefer, A. Bernstein, and A. Locher. Adding Data Mining Support to SPARQL via Statistical Relational Learning Methods. In *Proceedings of the 5th European Semantic Web Conference (ESWC)*, volume 5021 of *Lecture Notes in Computer Science*, pages 478–492. Springer-Verlag Berlin Heidelberg, 2008. to appear.

[4] C. Martenson, A. Horndahl, and Z. Kabir. An ontology-based adaptive reporting tool. In *roceedings of the Sixth International Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS 2011)*, 2011.

[5] R. Oldakowski and C. Bizer. Semmf: A framework for calculating semantic similarity of objects represented as rdf graphs. *Poster at the 4th International Semantic Web Conference (ISWC 2005)*, 2005.