Towards Collaborative Knowledge Engineering for Improving Local Safety in Urban Environment

Antoni Ligęza and **Weronika T. Adrian** and **Przemysław Ciężkowski**[1]

Abstract. Web systems supporting collaborative knowledge engineering have attracted much attention recently. By using social software techniques and attractive yet simple user interface, the motivation of users increases and the process can be significantly improved. The willingness of community to invest their time as well as mutual encouragement can be achieved when users are convinced that their contribution is important and useful. We propose a social platform called Social Threat Monitor (STM) aimed at improving safety of local communities in urban environment. The main assumption of the system is the collaboration of users to build and maintain a knowledge base about threats in their neighborhood. Knowledge gathered in the system can be used by the citizens as well as local authorities and police. The system supports collaborative knowledge engineering and management using semantic methods and a GIS component.

1 Introduction

Web-based information systems are used for gathering, storing and processing diversified information for various purposes. In Web 2.0. era, users can actively participate in building such systems. Projects such as Wikipedia have shown that collaborative knowledge acquisition (KA) and management (KM) can be successful if people see the importance of the project and the KA process is relatively easy. Mechanisms such as voting, commenting and discussions increase the possibility of building reliable and useful knowledge bases. Semantic technologies enable adding metadata to regular content and facilitate automatic knowledge extraction and processing [5].

In this paper, we present a Web-based system for collaborative knowledge acquisition and management. It is a thematic portal which aims to gather knowledge about threats of various kinds in local environment. This information may be used by citizens as warnings and by local police as notifications. The system is being developed within the INDECT project: ”Intelligent information system supporting observation, searching and detection for security of citizens in urban environment” [2]. The original contribution of this paper consists in presenting the collaborative knowledge engineering (KE) possibilities including knowledge exchange with external sources with use of a dedicated Application Programming Interface (API).

The paper is organized as follows: In Sect. 2 the motivation for this research is given. Sect. 3 provides an overview of the system functionality, user interface and implementation. Mechanisms applied to facilitate KE with the system are discussed in Sect. 4. The API of the system is presented in Sect. 5. Related work is outlined in Sect. 6, followed by a summary in Sect. 7 and future work in Sect. 8.

[1] AGH University of Science and Technology, Poland, email: {ligeza,wta}@agh.edu.pl

[2] See `http://indect-project.eu`.

2 Motivation

The aim of the Task 4.6. of the INDECT project is to develop a system for distributed knowledge acquisition and management with GIS [9] integration. The research is motivated by a hypothesis that local communities can effectively collaborate to build a useful knowledge base about threats in the neighborhood that can be used by both the citizens and local services or police. A system promoting collaborative knowledge acquisition and management should improve the communication between the citizens and the services, encourage cooperation within the community and thereby improve the local safety.

A social software platform facilitate collaborative knowledge engineering in an unintrusive way. Pieces of information shared by different persons are insignificant alone, but connected make a rich diversified picture. In popular social software platforms, people build personal knowledge bases half-consciously by acknowledging things and events shared by friends or ”followed” people. We want to leverage this dynamics and develop a system that would provide useful information while seamlessly integrating with daily life.

Within the INDECT project, several prototypes have been developed [3, 8], each of which constitute a information silo Web-based application. We claim that it is necessary to extend the existing prototypes to be more flexible and better adapted to knowledge interoperability. Knowledge gathered in the system should be easily exchanged with other applications that can use its data to process it in an arbitrary way (custom notifications, aggregation, statistics).

3 System Overview

The main goal of the system is to serve as a distributed knowledge acquisition system for data, information and knowledge provided by citizens, as well as to enable knowledge management and exchange. Principles and a conceptual model of the system have been described in [4]. The general idea of the system can be observed in Figure 1. The input data, in general, may be composed of: a text description of a threat, its spatial location, and multimedia documentation. The data, stored in a relational database equipped with spatial features, should be presented to the audience in a combined visual and textual form. The system provides means for searching, filtering, aggregation and grouping information for users, according to their preferred form and level of detail. The threats can be presented in a convenient and transparent way as icons on the map, in reports or notifications.

To enhance the automated knowledge processing of the system, semantic technologies for GIS were analyzed and discussed in [6]. The semantic research thread led to the development of a prototype described in [8] which investigates the integration issues of databases and ontologies. In the ontology, the general categories of threats were stored, whereas in the database the actual data about selected areas in

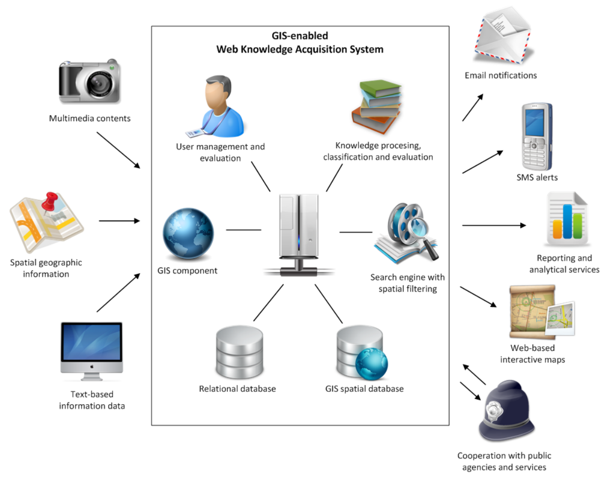

GIS-enabled
Web Knowledge Acquisition System
Multimedia contents
User management and evaluation
Knowledge procesing, classification and evaluation
Email notifications
SMS alerts
Spatial geographic information
GIS component
Search engine with spatial filtering
Reporting and analytical services
Web-based interactive maps
Text-based information data
Relational database
GIS spatial database
Cooperation with public agencies and services

Figure 1. Conceptual model of the system [3].

particular time were located. This prototype provided interesting insights and ideas for future investigations. However, for the INDECT purposes, more lightweight semantics and reasoning has been chosen. Three systems, referenced in [3] use lightweight reasoning and metadata annotations of threat such as simple tags. In the newest prototype [2], codenamed Social Threat Monitor (STM), only basic semantics is added with use of tags and categories (see Section 4). Summary of improvements with respect to the previous implementations can be found in [1].

The following groups of users are defined in the system:

Guest is a user with the anonymous web account. He is able to use basic features of the application. In order to gain more privileges, a guest need to register and log in into the system.

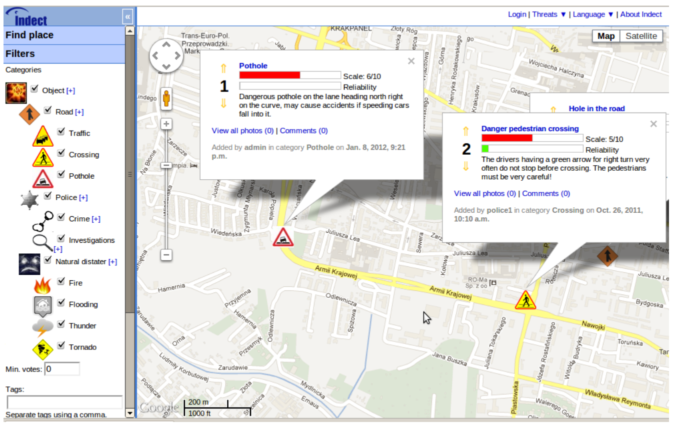
Member is a user with registered account in the system. With this account user can add threats, manage his own threats, comment and vote threats of other users and edit his profile.

Services User is a special account with features helping threats monitoring.

Moderator is a user with full access to threat records, able to ban users.

Administrator is a user with full access to the application.

Main part of application is the map (see Fig. 2) that covers all space user have and resize immediately when needed. The interac-



Pothole
1
Scale: 6/10
Reliability
Dangerous pothole on the lane heading north right on the curve, may cause accidents if speeding cars fall into it.
View all photos (0) | Comments (0)
Added by admin in category Pothole on Jan. 8, 2012, 9:21 p.m.

Danger pedestrian crossing
2
Scale: 5/10
Reliability
The drivers having a green arrow for right turn very often do not stop before crossing. The pedestrians must be very careful!
View all photos (0) | Comments (0)
Added by police1 in category Crossing on Oct. 26, 2011, 10:10 a.m.

Figure 2. User Interface of the Social Threat Monitor.

tive map implemented with use of AJAX technology – Asynchronous JavaScript and XML – provides easy and quick reloading as few times as possible. Because the URL does not change when only partial reloading is done, the identification of the state of the map (visible location or threat) is done with *hashing* (every time map position is changed or other action is performed, the hash changes).

The map have two editing modes: for browsing and adding threats. Mouse events are treated adequately to the mode. The map locates the user's position, but it can be moved (with mouse or keyboard) and zoomed (with mouse scroll or scroll on the left on the map).

Users browsing the system can immediately view short threat details (votes, one picture if available, number of comments and quick edit link). Voting is available only for registered users, but anonymous are able to see how many votes the threat got. Full gallery, list of tags and all comments are available on new page for each threat.

Top menu toolbar allows the user to toggle options of the login panel. If a user is already logged in, the login link is replaced with his account name and five options: 1) Profile (where the user can change account settings), 2) Logout, 3) Threats (where the user can list all threats, last added threats, top reliability threats and most dangerous threats), 4) Language (which allows user to change site language) and 5) About Indect – which is a link to the INDECT project page[3].

Most of the functionality is available form the left menu: (1) adding threats, including selecting area for a new threat, (2) browsing map by location, and (3) searching for threats using various filters. Each section of the left menu can be shown or hide.

The system has been implemented using widely-accepted, cost-free Web technologies: HTML, CSS, JavaScript, jQuery, Google Maps API version 3 and the Django framework (for details see [2]). It has been deployed on a dedicated server and is available at: `http://vigil.ia.agh.edu.pl`. In the wiki system there is a short description of the system, as well as user and admin manual.

4 Towards Collaborative Knowledge Engineering

Semantic annotations for the threats The basic piece of knowledge in the system represents a single *threat*. Each threat may be described using a set of attributes, such as: *geometry* – the information about the shape and the location of the danger, *name*, *category* – each danger is assigned to one category, *comments* – all logged in users may post their comment on the danger info, *severity* – a number telling how severe the danger is, *reliability* – a number telling to what degree the information is reliable, *photo gallery* – a relation to an object being a set of pictures illustrating the threat, *date added* – when the information has been added into the system, *modification date* – when the information has been last modified, and *tags* – in order to search for interesting information.

Tags Tags are non-hierarchical keyword or terms that describe an object, specifically a threat. One threat can be assigned several tags. They help searching and categorizing threats. Threats are tagged while adding by user. Well tagged threats are more reliable for other users and usually get more votes.

Categories Categories constitute a hierarchical way of describing threats. One threat has one category, but categories can contain many threats. Categories are organized into a tree structure. The root of the categories is not visible on site. There is no children limit for

[3] `http://www.indect-project.eu/`

categories. As relational databases are not designed for storing hierarchical data, retrieving category and its all parents with use of procedural SQL is complicated. Also, fetching the whole tree requires additional operations after database query execution. To solve this problems in STM, Nested Set Model was implemented through `django-mptt` module. It provides an efficient way to retrieve categories from database. Modifying the tree is more complicated, thus slower, but it is only possible for admin user and rarely executed.

User Groups Except for categories for threats, the users can also be grouped. Groups allows to publish threats for specified users. When adding a threat, a user (who is at least in Services group) can decide if the threat will be public or visible only for selected groups. This solution allows special groups have their own threats that will never be published for all system users.

An exemplary use-case has been presented in [1]. In this use-case, three police departments cooperate on an investigation. However, each group has their own sub-investigation and landmarks important to these investigations can be marked on map and visible only to selected users.

5 Towards Interoperability: the System API

Parts or the system data can be imported and exported to JSON, XML and YAML formats. In order to enable export of knowledge for further custom processing and import from another knowledge base, a simple Application Programming Interface (API) has been developed. Standarized knowledge representation using *attribute-value* pairs describing threats allows for using the system knowledge in various semantic applications (where triples of the form: *object-attribute-value* or *subject-predicate-object* are used). External systems can communicate with STM and use it as a web service. The API is available over HTTP protocol: `http://application.url/api/method_name/`.

API access and exchange format Part of the functionality of the Social Threat Monitor is available for external applications without authorization. For instance, by preparing an appropriate request conforming to the system API one can get all the threats defined for a given location or filter. In order to use the whole functionality, including adding threats to the system, the application must be authorized. Its user must be defined in the STM and assigned to the `API` group.

The API uses POST requests and HTTP cookies. All responses are in JSON format. Each successful request returns HTTP 200 header and the 400 header is returned, if a method does not exist. Moreover, if an anonymous user wants to access a method that requires authorization, a HTTP 200 header with appropriate content is returned.

Methods The system API provides three basic methods: 1) logging in, 2) adding a new threat, and 3) retrieving existing threats. Each method accepts arguments that must be sent using POST or COOKIES. Below, the methods with required (marked with an asterisk "*") and optional parameters are presented:

1. Method **login**:

   Description: A method allowing to log into the API.

   Parameters:

   - POST:
     - *username:`string` – user's name.
     - *password:`string` – user's password.

2. Method **add**:

   Description: A method allowing to add a threat.

   Parameters:

   - POST
     - *title:`string`
     - *description:`string`
     - *latitude:`float`
     - *longitude:`float`
     - *category:`int`
     - *scale:`int` from range $[1, 10]$ (severity of the threat)
     - *date_end:`int` from range $[1, 3]$ (the number indicates how many months a threat is active)
     - *tags:`string` – tags separated with comma
     - groups:`array(int)` – group ids for whom danger will be shown (empty for ”all groups”)
   - COOKIES
     - *sessionid:`string` – session id returned in login method

3. Method **threats**:

   Description: A method allowing to get filtered list of threats.

   Parameters:

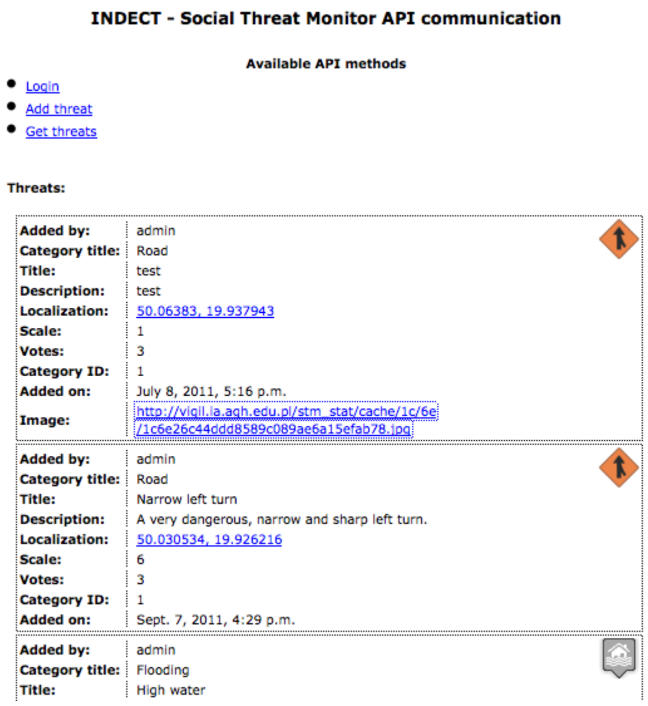
   - POST
     - *polygon: `string` – string of coordinates that define the area of interest, for example: `lng1 lat1, lng1 lat2, lng2 lat2, lng2 lat1, lng1 lat1`
     - votes: `int` – minimal votes number.
     - images: `int` from set $\{0, 1\}$ where 1 selectes threats only with photos.
     - category: `int` – category id.
     - tags: `string` – tags separated by commas.
     - scale: `int` from set $\{1, 4, 7, 10\}$ indicating threats scale.
     - date: `string` from set $\{12h, 24h, week\}$.
     - date_start: `datetime` – threats added after date.
     - date_stop: `datetime` – threats ending before date.
     - groups: `array(int)` – threats for groups.

   An example response:

Listing 1. Response to **threats** method on success.

```
{"threats": [
  {"category": 2,
    "user__username": "admin",
    "votes": 6,
    "scale": 5,
    "description": "Threat␣description",
    "point": {"latitude": 50.087389,
             "longitude": 19.891606},
    "title": "Crime",
    "category__img": "http://url/remont.png",
    "comments": 5,
    "date_add": "3␣kwietnia␣2011␣22:57:34",
    "points": 2,
    "category__title": "Crimes",
    "images": 7,
    "image__img": "http://url/name.jpg",
    "id": 1},
  {another_threat},
  {another_threat}]}
```

Demo implementation An example system using API has been developed and is available for testing at: `http://home.agh.edu.pl/kk/stm`. It has been implemented in PHP and uses cURL library. The library allows for connecting to and communicating with various types of servers and different protocols. An example screenshot showing STM response is presented in Figure 3.



INDECT - Social Threat Monitor API communication

Available API methods

- Login
- Add threat
- Get threats

Threats:

Added by:	admin
Category title:	Road
Title:	test
Description:	test
Localization:	50.06383, 19.937943
Scale:	1
Votes:	3
Category ID:	1
Added on:	July 8, 2011, 5:16 p.m.
Image:	http://vigil.ia.agh.edu.pl/stm_stat/cache/1c/6e/1c6e26c44ddd8589c089ae6a15efab78.jpg

Added by:	admin
Category title:	Road
Title:	Narrow left turn
Description:	A very dangerous, narrow and sharp left turn.
Localization:	50.030534, 19.926216
Scale:	6
Votes:	3
Category ID:	1
Added on:	Sept. 7, 2011, 4:29 p.m.

Added by:	admin
Category title:	Flooding
Title:	High water

Figure 3. API demo: A threat list returned upon request.

6 Related Work

Crime Mapping systems were originally a class of systems that map, visualize and analyze crime incident patterns using Geographic Information Systems (GIS). However, the name has been later extended to incorporate all applications that aid in improving the public safety. This include natural disasters monitoring systems, often designed for specific regions, which scope of functionalities is limited to the specific types of disasters that are most common and dangerous in those regions, systems monitoring threats on the roads and crime monitoring systems. A detailed survey of existing crime mapping systems is given in [10]. To the best of our knowledge, none of existing system works as a social platform supporting collaboration (voting, comments and collaborative evaluation of information).

7 Summary

One of the tasks within the INDECT Project is the development of a Web-based system for knowledge acquisition and management. Once the main assumptions and requirements were defined, the development has been done iteratively. Semantic description, categories and tags constitute the basis for further development of intelligent information processing and knowledge management. System API allows for easy integration with other applications and facilitate knowledge exchange and integration from various sources.

8 Future Work

The approach to semantics representation now used in the STM can be extended. Currently, only basic semantics is represented with use of tags and categories. They are closer to the model of folksonomies, where users provide custom tags that can be a foundation for a simple hierarchy of categories. A possible direction of future work is to refactor the hierarchy currently existing in STM with the use of a selected OWL 2.0 profile. All of the important relations should be identified and formalized. This will allow for having a complete formal model of the threat ontology. It is also planned to work on the rule-based engine [7] to manage and customize output channels.

Although the system works in a regular Web browser and thus can be accessed from any mobile device that has a browser, further adaptation for smartphones is planned. In particular, the system should use the GPS embedded in mobile devices to facilitate adding threats.

ACKNOWLEDGEMENTS

The research presented in this paper is carried out within the EU FP7 INDECT Project: "Intelligent information system supporting observation, searching and detection for security of citizens in urban environment" (`http://indect-project.eu`).

References

[1] Weronika T. Adrian, Ciężkowski, Krzysztof Kaczor, Antoni Ligęza, and Grzegorz J. Nalepa, 'Web-based knowledge acquisition and management system supporting collaboration for improving safety in urban environment', in *Multimedia Communications, Services and Security*, eds., Andrzej Dziech and Andrzej Czyżewski, volume 287 of *Communications in Computer and Information Science*, 1–12, Springer Berlin Heidelberg, (2012).

[2] Przemysław Ciężkowski, *Functionality Analysis and Design and Implementation of User Interface for Threats Enregistration in Internet System*, Master's thesis, AGH University of Science and Technology, 2011.

[3] Antoni Ligęza, Weronika T. Adrian, Sebastian Ernst, Grzegorz J. Nalepa, Marcin Szpyrka, Michał Czapko, Paweł Grzesiak, and Marcin Krzych, 'Prototypes of a web system for citizen provided information, automatic knowledge extraction, knowledge management and gis integration', in *Multimedia Communications, Services and Security*, eds., Andrzej Dziech and Andrzej Czyżewski, volume 149 of *Communications in Computer and Information Science*, 268–276, Springer Berlin Heidelberg, (2011).

[4] Antoni Ligęza, Sebastian Ernst, Grzegorz J. Nalepa, and Marcin Szpyrka, 'A conceptual model for web knowledge acquisition system with GIS component', *Automatyka: półrocznik Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie*, **13**(2), 421–428, (2009).

[5] Grzegorz J. Nalepa, 'Collective knowledge engineering with semantic wikis', *Journal of Universal Computer Science*, **16**(7), 1006–1023, (2010).

[6] Grzegorz J. Nalepa and Weronika T. Furmańska, 'Review of semantic web technologies for GIS', *Automatyka: półrocznik Akademii Górniczo-Hutniczej im. Stanisława Staszica w Krakowie*, **13**(2), 485–492, (2009).

[7] Grzegorz J. Nalepa and Antoni Ligęza, 'HeKatE methodology, hybrid engineering of intelligent systems', *International Journal of Applied Mathematics and Computer Science*, **20**(1), 35–53, (2010).

[8] Jarosław Waliszko, Weronika T. Adrian, and Antoni Ligęza, 'Traffic danger ontology for citizen safety web system', in *Multimedia Communications, Services and Security*, eds., Andrzej Dziech and Andrzej Czyżewski, volume 149 of *Communications in Computer and Information Science*, 165–173, Springer Berlin Heidelberg, (2011).

[9] *The Handbook of Geographic Information Science*, eds., John P. Wilson and A. Stewart Fotheringham, Blackwell Publishin Ltd, 2008.

[10] Maciej Żywioł, *Analysis and Evaluation of Crime Mapping Systems*, Bachelor's thesis, AGH University of Science and Technology, 2012.