$crowd^{SA}$ — Towards Adaptive and Situation-Driven Crowd-Sensing for Disaster Situation Awareness

Andrea Salfinger, Werner Retschitzegger, Wieland Schwinger
Dept. of Cooperative Information Systems
Johannes Kepler University Linz
Altenbergerstr. 69
4040 Linz, Austria
{andrea.salfinger, werner.retschitzegger, wieland.schwinger}@cis.jku.at

Birgit Pröll
Inst. for Application Oriented Knowledge Processing
Johannes Kepler University Linz
Altenbergerstr. 69
4040 Linz, Austria
bproell@faw.jku.at

***Abstract*—Disasters pose severe challenges on emergency responders, who need to appropriately interpret the situational picture and take adequate actions in order to save human lives. Whereas Information Fusion (IF) systems have proven their capability of supporting human operators in rapidly gaining *Situation Awareness* (SAW) in control center domains, *disaster management* presents novel challenges: Due to the *unpredictability*, *uniqueness* and *large-scale dimensions* of disasters, their situational pictures typically cannot be extensively captured by sensors — a substantial amount of situational information is delivered by human observers. The ubiquitous availability of social media on mobile devices enables humans to act as *crowd sensors*, as valuable crisis information can be broadcast over social media channels. Although various systems have been proposed which successfully demonstrate that such crowd-sensed information can be exploited for disaster management, current systems mostly lack means for automated reasoning on these information, as well as an integration with structured data obtained from other sensors. Therefore, in the present work we provide a first attempt towards comprehensively integrating social media-based crowd-sensing in SAW systems: We contribute an architecture on an adaptive SAW framework exploiting both, traditionally sensed data as well as unstructured social media content, and present our initial solutions based on real-world case studies.**

I. Introduction

Situation Awareness & Disaster Management. Natural and man-made disasters pose severe challenges on emergency responders, who need to appropriately interpret the situational picture and immediately take adequate rescue actions in order to save human lives. To support human operators in rapidly gaining *Situation Awareness* (SAW) in the light of massive amounts of data, systems capable of automated situation assessment (SA) have been proposed. Whereas such Information Fusion (IF) systems have already shown their usefulness in a range of control center domains mainly operating on hard-sensor data (e. g., road-traffic and air-traffic control, maritime monitoring), the domain of *disaster management* presents novel challenges: Unlike domains that can be comprehensively monitored by hardware sensors, disasters typically cannot be extensively captured by such sensors due to their *unpredictability*, *uniqueness* and *large-scale dimensions* [1] — thus, a substantial amount of situational information is delivered by on-the-ground human observers.

Crowd-Sensing. However, the ubiquitous availability of social media (SM) on mobile devices enables humans to act as *crowd sensors* (or citizen sensors [2]), as valuable crisis information can be broadcast over SM channels (e. g., as studied in [3]–[6]). Although primarily intended for human communication, various systems have been proposed which successfully demonstrate that such information can also be exploited in disaster management systems (e. g., [7]–[15]). However, current systems mostly lack means for automated reasoning on SM content, as well as an integration with structured data obtained from other sensors, and provide limited self-adaptivity, semantic processing, identification of sparse situational updates and incorporation of situative context.

Contributions. Therefore, in the present work we provide a first attempt towards comprehensively integrating SM-based crowd-sensing in automated SAW systems: We contribute an architecture on a SAW framework exploiting both, traditionally sensed data as well as unstructured SM content, and present our initial solutions for the challenges of SM-sensing, namely (i) coping with the dynamics of SM by providing an adaptive crowd-sensing level [C.1][1], (ii) a semantic analysis of textual content [C.2], (iii) identifying sparse *situational update information* posted by on-the-ground observers [C.3], and (iv) studying means for employing additional context by exploiting already assessed or projected situations [C.4].

Structure of the Paper. In the next section, we discuss the state-of-the-art of SAW systems incorporating crowd-sensed content, and elaborate on open issues. In Sec. III, we subsequently propose the architecture of $crowd^{SA}$, a SAW system for disaster management which attempts at providing solutions towards the open issues we identified in Sec. II. Finally, we provide an outlook on future work in Sec. IV.

II. Related Work

Based on case studies revealing the potential of SM for supporting disaster management (e. g., [3]–[6]), several efforts

[1]Note that these enumerated abbreviations will be used in the following sections to refer to the corresponding challenge.

have been undertaken to make these information accessible to emergency managers and operators. To assess how current crowd-sensing systems for disaster management comply with the requirements on full-fledged SAW systems (e.g., [16]), we conducted a survey evaluating state-of-the-art systems with respect to (w.r.t.) the IF levels these address (as specified in the JDL data fusion model [17]): Sensing data from the observed environment (JDL Level 0), assessing objects from these measurements (JDL Level 1), assessing the overall ongoing situations (JDL Level 2), projecting these situations' development and impact (JDL Level 3), and furthermore, resource management or process refinement (JDL Level 4) and user refinement (JDL Level 5). In our evaluation, we contrasted the following approaches [18]: HADRian [7], ESA [8], Twitris [9], Twitcident [10], SensePlace2 [11], CrisisTracker [12], TweetTracker [13], Toretter [14], and CIACM [15]. Reviewed from the perspective of a comprehensive SAW architecture stretching across all IF levels, it became apparent that current systems often lack means for automated SA (which is, except for HADRian [7], deferred to the human operator), thus do not support JDL levels 2+, and mainly rely on human expertise and interaction. Although providing valuable first steps towards crowd-sensing enhanced SAW systems, these systems are designed to be steered by a human operator and her ad-hoc information needs and insights, but provide limited self-adaptivity (towards emerging SM trends), thus, do not address [C.1], and system-based, continuous monitoring functionality implementing situation assessment and projection (which is not targeted in any of the above mentioned systems). Besides that, many systems base on clustering approaches to infer real-world events from SM content [8]–[10], [12] or probabilistic event detection [9], [14], [15], thus, summarize *frequently* posted information, which is assumed to represent a certain degree of confidence. However, actually sparse *situational update information* [19] may be dominated by general news and comments, and therefore may not be brought to the operator's attention, thus, [C.3] is not addressed. This is aggravated by the fact that most current systems do not employ a natural language processing approach for an in-depth analysis of the actual semantics comprised in the messages, but pursue a bag-of-words approach (compare textual similarity based on overlapping words), nor include situative context, thereby do not address [C.2] and [C.4].

III. A Framework for Engineering Crowd-Sensing Enhanced SAW Systems

In this section, we introduce $crowd^{SA}$ [20][2], a SAW system capable of exploiting both, data delivered from hardware sensors as employed by control centers, as well as information retrieved from SM, i.e., *crowd-sensed* information. In contrast to existing crowd-sensing-based SAW systems (cf. Sec. II), $crowd^{SA}$ specifically aims at providing means for a *situation-driven sensing and perception configuration* by implementing feedback-loops between the different processing levels, which adapt the lower-level IF steps and provide additional context for the retrieval and interpretation of SM content. In order to realize this functionality, we propose concepts for the following issues that need to be overcome: (i) coping with the dynamics of SM by providing an *adaptive crowd-sensing* level [C.1], (ii) employing a *semantic analysis* of textual content based on domain ontologies in order to infer the reported events and match them to ontological concepts that can be used for automated reasoning (SA) [C.2], (iii) specifically *seeking sparse situational update information* posted by on-the-ground observers by proposing an aggregation-segregation-based approach [C.3], and (iv) employing additional context in order to infer the semantics of SM content by exploiting already assessed or projected situations [C.4].

For realizing the core SAW system, we base on our previous implementations of SAW systems for control centers, BeAware! [21], [22] and CSI[3] (*Collaborative Situation Awareness in Road Traffic Control*). The conceptual architecture of the crowd-sensing level is based on insights and lessons learned from our currently on-going implementation of the devised crowd-sensing level, and initial case studies on real-world Twitter[4] data sets (cf. Table I). In particular, we will illustrate the processing outlined in Fig. 1 based on three different disaster events (and their aftermath) happening between Aug., 9th and 15th, 2014, which are reflected in our collected data sets, notably (i) hurricane-events affecting the Hawaiian islands[5] (cf. Table I), (ii) the flooding caused by the remnants of hurricane Bertha in UK[6], and (iii) a severe flooding event on Long Island, New York[7]. Note that we will refer to individual tweets shown in this Fig. based on their assigned reference, denoted by [T.x], whereby x corresponds to a sequential digit. The different parts and components of Fig. 1 are referenced and explained in the forthcoming sections wherever appropriate (the functional blocks are referenced by encircled numbers), although the principal reading direction of Fig. 1 is from bottom to top. We start our discussion of the processing layers summarized in Fig. 1 by illustrating the functionality of the Core SAW system, before we elaborate on the challenges and solutions w.r.t. the incorporation of additional JDL L0 and L1 components realizing *crowd-sensing*, i.e., the retrieval and fusion of content from SM.

A. CSI — the Core SAW System

In control center monitoring tasks, such as road traffic control or disaster management, human operators monitor their environment, observed over various sensors, for the occurrence of various event constellations that require specific counteractions, which we term *situations*, such as (i) the formation of a traffic jam in an area of dense fog, requiring the operator to display warnings on variable message signs to alert approaching drivers unaware of the jam, or, (ii) power outages in specific areas, which may affect sensitive infrastructure that requires specific action (such as hospitals, which may eventually require evacuations if the time required for restoring power exceeds their emergency generators' capacity).

As situations in these domains may be composed of heterogeneous types of objects, and could either correspond to frequently recurring situations (e. g., fusing traffic jams), or denote rather seldom and unique situations (as typical for large-scale disasters), we pursue a *knowledge-based* approach for SA, as also proposed in [23]: Event constellations that should be detected by the system are described by a set of rules. SA conforms to matching data observed from the environment against these rules, whereby a matched rule corresponds to the detection of a specific real-world situation instance.

1 **Configuration** CSI requires domain experts to specify templates describing the situations of interest, so-called *Situation Evolution*

[2] crowdsa.situation-awareness.net

[3] csi.situation-awareness.net

[4] Therefore, from now on we will use the terms *social media message* and its Twitter-specific equivalent *tweet* interchangeably.

[5] http://www.latimes.com/nation/nationnow/la-na-nn-hawaii-storm-iselle-juliio-20140808-story.html

[6] http://www.bbc.com/news/uk-scotland-28739164

[7] http://newyork.cbslocal.com/2014/08/13/flash-flood-watches-warnings-in-effect-as-heavy-rain-drenches-parts-of-tri-state/

Data Set	Purpose/Event	Time Period	Service	Remarks
GeneralDisasters	general Twitter stream monitoring for English disaster-related keywords (Typhoon, Hurricane, Flooding, #Typhoon, #storm, #typhoon, #flood, flood, spring tide, windstorm, disaster)	07/31/2014 — 10/31/2014	Twitter Streaming API & Twitter4J	$> 7.3 \times 10^6$ tweets
HawaiiHurricanes	Hurricanes *Iselle* and *Julio* and the tropical storm *Genevieve* passing Hawaii (by filtering tweets according to the keywords: Hurricane, #HurricaneIselle, #HurricanePrep, #updatehurricaneiselle, #hiwx, #HIGov, Iselle, #Genevieve, #Iselle, #Julio, #HIWX, #HIWx)	08/09/2014 — 08/21/2014	Twitter Streaming API & Twitter4J	∼ 212.600 tweets

Table I
OVERVIEW OF COLLECTED REAL-WORLD DATA SETS.

Types (SETs) [24], which are formulated in terms of a suitable ***Domain Ontology***. Due to performance considerations, we decided to favor an object-relational implementation in CSI, as opposed to the semantic web technologies employed in our previous SAW framework BeAware [21] (namely RDF, the graph database Allegrograph, Lisp and Prolog), motivated by our comparative performance evaluation of the two approaches in [22]. Therefore, we use a UML-based approach for ontology engineering [25], employing the model-driven development tool Visual Paradigm[8] as ***OntologyDesigner***. Our SAW Core Ontology is based on [26], whereby new application domains can be incorporated easily by simply inheriting from our base classes, such as *Object*, *Situation* and *Action*. After ontology engineering, the corresponding database schemes, Data Access Layers, and persistence mappings can be automatically generated by Visual Paradigm. For our current implementation, we employ a postgreSQL database with the spatial extension postGIS, Java for implementing the framework logic, the OR-mapping framework Hibernate[9], and the business rules platform JBoss Drools Rules[10].

2 **SET Modeling** After configuring the SAW framework towards a specific application domain, domain experts need to populate the ***Situation Evolution Type Knowledge Base*** (KB) with the sought-after SETs. A SET models the different potential evolutionary states of a situation, i.e., allows to track a crisis situation from its emergence (e. g., the situation *Hurricane threatens inhabited area* is triggered by the formation of a hurricane moving towards inhabited landmass) through its climax (e. g., the hurricane makes landfall and causes damage, such as power outages and flooded roads) to its clearance (e. g., power is restored). Thus, it corresponds to a Finite State Machine (FSM) describing the different *states* of a situation (termed Situation State Types, short SSTs), which correspond to a set of *Event* and *Object Types* (e. g., *PowerOutage*, *City*) in specific relations (e. g., the spatial relations *Overlapping*, *Close*, the temporal relation *Before*), and the possible transitions between these states [24], [27]. In order to facilitate SET specification and mitigate the knowledge acquisition bottleneck, we devised ***SEM^2 Suite***, a tool suite supporting an interactive and incremental specification of such SETs [28]. A screenshot of a SET specified in ***SEM^2 Suite*** is shown in Fig. 2, depicting a SET capturing potential SSTs encountered in expectation of, during and after a hurricane disaster (exemplarily showing also the specification of the SST "PowerOutageInCity", and an aggregate SST based thereupon, "PowerOutageInArea"). In order to enable automated ***Situation Assessment***, ***SEM^2 Suite*** compiles each of these SSTs to a rule supplied to the ***Situation State Assessor***'s rule engine, and stores the specified SETs in the the ***Situation Evolution Type KB***, which is used at runtime by the ***Situation Evolution Assessor*** to reason upon their defined FSMs to infer evolving situations.

Situation Assessment At runtime, SA is performed in the following fashion, implementing the JDL data fusion model [17]:

JDL L0. The 3 *Sensing Level* retrieves various observations from the environment (e.g., measurements obtained from wind or humidity sensors, or satellite images).

[8]http://www.visual-paradigm.com
[9]http://hibernate.org
[10]http://www.drools.org

JDL L1. The 4 *Perception Level* performs object assessment, i.e., infers the monitored real-world ***Objects*** from these measurements (e.g., reconstructs a hurricane's location from satellite data).

JDL L2. The 5 *Comprehension Level* infers the overall situational picture based on analyzing the objects interrelations, which is implemented in a two-tier fashion: First, the ***Situation State Assessor*** aims at detecting currently on-going situations: Matched rules (corresponding to a specific SST) trigger the creation of a ***Situation State*** instance, i.e., a snapshot of a real-world situation. Second, the ***Situation Evolution Assessor*** performs situation evolution tracking. Based on reasoning on the SETs stored in the ***Situation Evolution Type KB***, the currently detected ***Situation States*** are compared with previously assessed situation instances, in order to infer whether the currently assessed situation snapshots correspond to novel situation instances or to an evolution of already detected situations [24].

JDL L3. In order to take the adequate counteractions, the 6 *Projection Level* aims at forecasting the encountered situations' development (e.g., if already several mudslides and high water levels have been reported and the weather forecast predicts prolonged rainfalls, the situation will likely escalate and thus demand preventive action, such as securing dams). Therefore, the ***Situation Evolution Predictor*** reasons upon currently on-going ***Situation Evolutions***, forecast data (e. g., weather forecasts), and potentially historic ***Situation Evolutions*** (i.e., historic situations are employed to compute the most probable situation evolution).

B. $crowd^{SA}$ — Incorporating Crowd-Sensing

Whereas in the hurricane scenario described above, some SSTs can be assessed from authority-specific hardware sensors (e. g., the detection of a hurricane formation, and the projection of its likely movement, can be accurately computed from various weather sensors and satellite sources), the assessment of actually encountered damage (as well as the determination of non-affected areas and needed resources) is largely based on human reports and observations. We will thus elaborate on how relevant observations can be retrieved from SM, and serve in the inference of the situational picture, i.e., we seek to fulfill the specified SETs with observations from SM.

7 **Crowd-Sensor Management [C.1] & [C.4] — Retrieving Potentially Relevant Messages.** *Authority sensors*, no matter whether delivering structured data (such as obtained from different kinds of hardware-sensors), or unstructured, i.e., free-form textual content (such as contained in protocols), are dedicated to record data related to the domain-tasks at hand. Contrastingly, the inclusion of crowd-sensed SM content introduces novel challenges to the *Sensing Level* (JDL L0) due to the *untargetedness* of SM: As SM users may chat about virtually any topic, actual content of potential relevance needs to be determined beforehand. To address the monitoring requirements of large-scale applications, the ***Crowd-Sensing Platform*** can be configured to run dedicated adapters for different SM platforms and their filtering components on an Apache Storm[11] cloud, allowing for distributed processing meeting real-time demands. Our implementation is currently focused on Twitter, providing two kinds of adapters: One for continuous monitoring over the Twitter Streaming API (using

[11]https://storm.apache.org

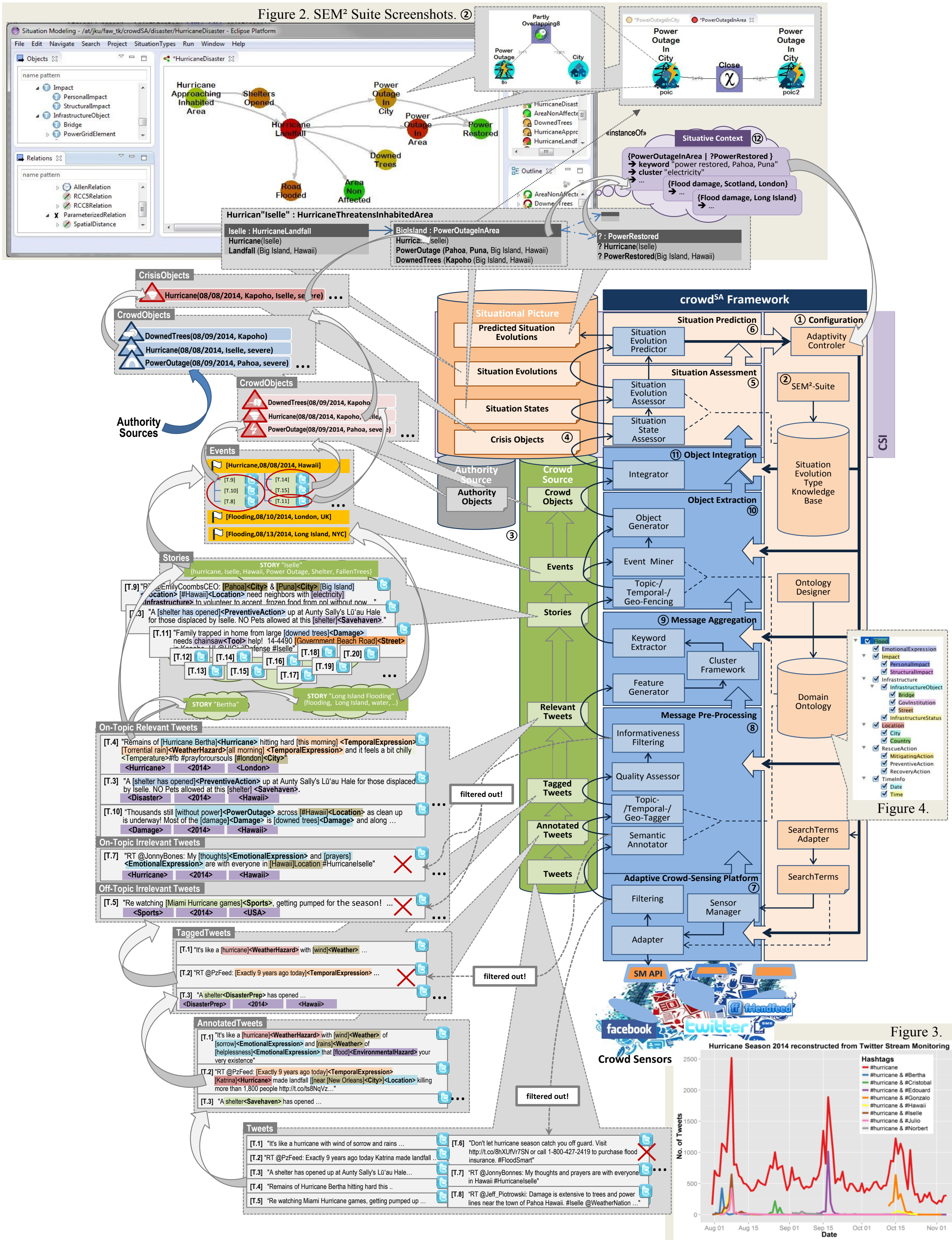

Figure 1. *crowd*^{SA} architecture and processing illustrated.

the library Twitter4J[12]), another one for post-event data gathering over the Twitter Search API. The adapters retrieve content matching specific ***Search Terms*** (comprising keywords and potentially an area of interest, i.e., a geographical boundary box), and are steered by the ***Sensor Manager***.

To address [C.1], i.e., the quickly changing nature of SM, a ***SearchTerms Adapter*** aims at performing adaptive keyword selection. As the analysis of our domain data sets revealed, crises typically trigger the adoption of crisis-specific hashtags: In the *HawaiiHurricanes* dataset, for instance, 4728 tweets contained the crisis-specific hashtag *#Iselle*, but not the general hashtag *#hurricane* (98% of these did not even comprise the word 'hurricane' at all). Such specifically marked situational update information may be missed by general disaster-related keywords (e.g., by just filtering for messages comprising "hurricane"). However, our analysis showed that such crisis-specific hashtags can be inferred from their emerging co-occurrence with general disaster-related keywords: Fig. 3 plots the tweet intensity on hashtag-filtered subsets from the *GeneralDisasters* data set over time, whereby this plot actually allows to reconstruct the different hurricane events that occurred in 2014 (which matches the "ground truth", i.e., the records of the National Hurricane Center[13]): As peaks occur for tweets using the hashtag *#hurricane*, a smaller peak can be observed at the same time for tweets containing both, the general hashtag *#hurricane*, as well as a hashtag assembled from the name of the hurricane happening at that time (e. g., *#Iselle*). We could observe similar behavior in other data subsets on typhoons and other types of crisis (not shown here due to space constraints). Therefore, the ***SearchTerms Adapter*** seeks to detect such emerging hashtag correlations, upon which the ***Sensor Manager*** may spawn new adapter instances tracking for crisis-specific hashtags.

The ***Sensor Manager*** acts as the central interface for realizing both JDL L4 and L5 functionality, thereby addresses [C.4]: Based on a (partially) assessed or projected situation, the ***Situation Evolution Assessor*** or the ***Situation Evolution Predictor*** may seek to retrieve missing or additional information over SM channels (e. g., if a hurricane or storm occurred and power outages have been reported, the system actively searches for tweets reporting damage or seeking help from the affected area, as sketched in 12 in Fig. 1). Thus, these components specify a *situation profile* comprising the location and information types of interest (elaborating on the idea of the *initial incident profile* proposed in [10]). This request is supplied to the ***Sensor Manager***, which translates this *situation profile* to a corresponding ***Adapter*** keyword set. Besides that, based on the operator's current information need, she may wish to issue specific queries, which are also propagated to the ***Sensor Manager*** responsible for query expansion and configuring the appropriate Adapter instances.

Furthermore, the ***Crowd-Sensing Platform*** performs an initial quality ***Filtering*** (comprising Spam filtering, i.e., excluding the content of blacklisted users and sources), such as excluding (insurance) advertising tweets, for example [T.6].

8 Semantic Preprocessing [C.2] & [C.4] — Making Sense of Retrieved Messages. After storing the retrieved SM messages in a central ***Tweet Repository***, the next processing step comprises the ***Semantic Annotation Pipeline***, i.e., performs Natural Language Processing (NLP) on the actual message text. For this step, $crowd^{SA}$ employs the popular NLP software framework GATE [29], which allows for a custom configuration of various annotation pipelines, and its extension fine-tuned towards the linguistic peculiarities of Twitter content, TwitIE [30]. As opposed to open-domain Information Extraction (IE) tasks, for the application domain of disaster management, a priori knowledge of the different information types of interest can be employed, as has been demonstrated in [31] for the identification of seeker and supplier behavior in Twitter during crisis situations (termed *domain-dependent* analysis of message content). For the tasks of disaster management, the following categories of information are of primary interest [19]: (i) reports on the current state of infrastructure and (ii) on different types of hazards, and (iii) requests for and supplies of resources or help. In order to extract these information, we pursue an *ontology-based semantic annotation* approach [C.2], by employing a domain ontology for annotating information entities in tweets, such as infrastructure entities (e. g., mentions of bridges, buildings etc.), or natural phenomena (e. g., environmental hazards such as flooding or hurricanes). Our current implementation bases on and extends the flooding-specific ontology proposed in [32] (an excerpt of which is shown in Fig. 4), which we modeled in RDF, and is loaded into GATE during the annotation phase. Also general open-domain knowledge for annotation purposes from DBpedia[14] is integrable, for instance for the inference and annotation of proper nouns and proper names (e. g., the Everglades, Microsoft), by configuring the DBpedia Spotlight plugin[15] into the annotation pipeline. Our basic annotation pipeline consists of the classical NLP steps of tokenization, stopword removal, Part-of-Speech (POS) tagging, stemming, Named Entity Recognition and Classification (NERC) based on ontological and Gazetteer-based lookups, resulting in domain-grounded annotations.The mapping from plain text to ontological concepts allows to resolve syntagmatic relationships (e. g., synonyms) and language heterogeneity, an important issue in SM [33], and particularly crucial in disaster management [5]: To adapt the processing pipeline towards different languages, the text preprocessing components (stopword removal, POS tagging, stemming) need to be configured towards the language at hand. Furthermore, the ontological classes need to be annotated with textual labels of the corresponding language, whereas the overall annotation pipeline remains unchanged.

Upon this initial semantic interpretation, the overall *topic*, *time* and *location* of each tweet need to be determined, i.e., it needs to be inferred *what* has happened *where* and *when*, which may be supported by a *situative context* obtained from a feedback loop from the comprehension or projection level [C.4]. This *spatio-temporal-thematic* grounding is performed by three dedicated components, i.e., a ***Topic Tagger***, a ***Temporal Tagger***, and a ***Geo-Tagger***. The latter two are motivated by the fact that a tweet's metadata on its creation timestamp and its user's location need not necessarily correlate with its content. Therefore, the ***Temporal Tagger*** aims at resolving the tweet content's time span, i.e., performs *temporal grounding*, which is a key requirement in order to correctly interpret (and discard) messages such as [T.2].

The ***Geo-Tagger*** aims at anchoring a tweet to one or multiple geographical locations, i.e., performs *spatial grounding* of the reported event. Therefore, it potentially needs to incorporate the four different location types encountered in tweets, i.e., *User's Location Profile* (e. g., the user's home town specified in her profile), *User's Current Location* (if the tweet has been sent from a mobile device), *Locations in Text* (any location mentioned in a text, which could also be, e. g., 'London Press') and *Focused Locations* (locations mentioned in text that are indeed the locations of mentioned events) [34] and finally decide upon which location(s) the tweet should be mapped to. In [T.2], the ***Geo-Tagger*** should detect "near New Orleans" as focused location, and thus map the hurricane event to the area around New Orleans. All textually specified locations furthermore require *toponym resolution* or *geo-coding*, i.e., the mapping of the location name to the actual geographic positions (comprising latitude and longitude). For this task, $crowd^{SA}$ employs the GeoNames ontology and geo-coding service[16]. However, both *geo-* and *non-geo-ambiguity* (i.e., common words need to be distinguished from proper names, e.g., *Reading* may refer to a verb or a city in the UK), as well as the *ambiguity of location names* (e. g., Sydney may refer to a city in

[12]http://twitter4j.org

[13]http://www.nhc.noaa.gov/data/tcr/index.php?season=2014

[14]http://dbpedia.org

[15]https://github.com/jendarybak/GATE-DBpedia_Spotlight

[16]http://www.geonames.org

Australia or Canada) [35], require additional contextual information. Therefore, feedback-loops from the subsequent processing levels are used in order to provide additional context, thus addressing [C.4]: For instance, if a specific tweet in question is missing location mentions, but the *User's Location Profile* is set to New York, U.S., and a situation has been assessed that a severe flooding event is happening in New York, which matches that tweet's content, the tweet may be consequently annotated with the location New York.

Based upon NERC performed by the ***Semantic Annotator*** in the previous step, these semantic annotations are employed in the subsequent ***Informativeness Filtering***. Whereas previously proposed approaches for informativeness filtering predominantly suggest machine-learning based methods operating on pure text vectors (e. g., [36], [14]), we seek to consider the *semantics* of the tweet (in order to resolve synonyms, complex relations and negations, e.g., mapping the phrases "no electricity", "without power" [T.10], "w/o power", and "damage ... to ... power lines" [T.8] to the ontological concept *PowerOutage*) for classifying them according to the following categories:

— *off-topic*: Tweets which are not related to a disaster, but retrieved since matching the ***Adapter***'s keywords, which are used in a different semantic context, such as [T.1] and [T.5].

— *on-topic, irrelevant*: Tweets that are related to the disaster, but apparently do not contain on-the-ground or situational update information. These mostly correspond to emotionally focused tweets commenting on the disaster, which, however, may be of value for characterizing the evolutionary phase and severity of a crisis, such as [T.7].

— *on-topic, relevant*: This category comprises tweets which contain crisis-related information, such as [T.3-4], [T.8-11].

To assign a ***Semantically Annotated Tweet*** to one of these categories, the ***Quality Assessor*** aims at determining its *information content*: It computes dedicated quality metrics which aggregate over the encountered entity annotations, i.e., annotations of ontologically relevant information, such as mentions of infrastructure elements (e. g., bridges, buildings), crisis preparation, mitigation and recovery actions, and natural phenomena (e. g., flooding, hurricane). Thus, the mentioning of a location or infrastructure entity may increase a tweet's information content value, whereas emotional expressions (e. g., the phrases 'pray for', emoticons such as ";)"), for example, may decrease this value. This approach has been motivated by findings based on a manual inspection of large crisis data sets from Twitter reported in [19], where it has been concluded that tweets that comprise *situational update information*, and have been posted with the aim to broadcast these information, contain higher information content, and are more often marked with location information (e. g., [T.9]), than commentary tweets (e. g., [T.7]).

Therefore, we propose a dedicated ***QualityAssessor*** component, which comprises a ***QualityMetricCalculator*** that can be configured towards various ***MetricCalculationStrategies***. Whereas our initial experiments on this knowledge-based ***Informativeness Filtering*** yielded promising results w.r.t. the identification of disaster-situation relevant tweets, we currently experiment with incorporating suitable ***Quality Metrics*** and their thresholds for category assignment: For instance, the co-occurrence of temporal information, location information, disaster types and infrastructure or rescue action entities in a tweet could be attributed with larger weights. Furthermore, more specific location information, such as *Pahoa* and *Puna* (e.g., in [T.9]), should be attributed with larger weights than coarse-grained information such as *Hawaii*.

9 Message Aggregation [C.2] & [C.4] — Grouping Related Observations. After semantic-based ***Quality*** and ***Informativeness Filtering***, only the ***Relevant Annotated Tweets*** are retained for the further processing steps: In order to infer the underlying real-world events discussed in retrieved messages, i.e., perform JDL L1 *Object* or *Event Assessment*, similar messages will be grouped (following the assumption that the underlying real-world event will be sensed by multiple crowd-sensors observing the same event), which also follows the finding that SM information is mainly of use on an aggregate level [5]. Whereas most crowd-sensing approaches perform clustering based on text vectors, for instance employ the cosine-similarity between the tweets' word vectors (e. g., [12]), we plan to investigate on how we can incorporate features derived from semantic annotations in the clustering procedure [C.2]. Therefore, $crowd^{SA}$'s ***Message Aggregation*** component consists of a ***Feature Generator***, allowing for the specification of custom features, and a general ***Cluster Framework***, allowing to employ different cluster algorithms. Whereas the ***Cluster Framework*** simply returns sets of tweets that have been determined to be *similar* w.r.t. the defined features, i.e., should presumably comprise tweets discussing the same underlying real-world event, the ***Keyword Extractor*** aims at identifying common terms and keywords from this tweet set, resulting in a so-called ***Story*** (following the terminology proposed in [12]), i.e., a set of tweets presumably discussing the same topic, and keywords or entities that can be considered as descriptive of this tweet cluster.

10 Event Detection & Tracking [C.3] & [C.4] — Inferring Real-World Events From Grouped Observations. In order to finally infer the underlying *real-world* event that is presumably discussed within this tweet cluster, suitable *spatio-temporal-thematic* descriptors need to be extracted from this ***Story*** [37], i.e., the system needs to determine *what* has happened *where* and *when*. This is similar to what has been performed during the ***Message Pre-Processing*** phase on a per-message basis, but in this phase needs to be inferred from a set of tweets, which is performed by a ***Topic Fencing***, a ***Geo-Fencing*** and a ***Temporal Fencing*** component, respectively, ultimately yielding a spatio-temporal-thematic description of the inferred real-world ***Events***. From ***Stories*** shown in Fig. 1, for instance, the following real-world events can be inferred: Tweets mentioning the damage after Hurricane Iselle passed the Hawaiian islands, tweets grouping around the event of the tropical storm Bertha crossing UK, and tweets mentioning the flooding of Long Island.

In order to be interpretable for the ***Situation State Assessor***, these ***Event***-level descriptions need to be mapped to ***Object***s of the ***Domain Ontology***. Thus, the ***Object State Generator*** attempts to map events to the ***Domain Ontology***, and instantiate corresponding ***Object States***. Finally, the ***Object Evolution Detector*** needs to infer whether the detected ***Object State*** corresponds to a new, i.e., previously unseen object, or denotes the evolution of an existing object, finally yielding the inferred ***CrowdObjects***.

It is noteworthy to mention that a single ***Story*** may describe multiple ***Sub-Events***, characterized by specific subclusters on these subjects within the ***Story***, and thus trigger the instantiation of multiple ***Objects***. The stories depicted in Fig. 1, for instance, trigger the instantiation of the ***Objects*** *Power Outage* and *Fallen Trees* due to an ***Event*** *Hurricane* (hurricane Iselle in Hawaii), and the ***Objects*** *Torrential Rain* and *Flooding* due to another ***Event*** *Hurricane* (hurricane Bertha in UK). Ultimately, the correlation of evolving ***Events*** and ***Objects***, as triggered by disasters and discussed in ***Stories***, could potentially provide a means for *learning SETs* from the crowd, i.e., crowd-based knowledge acquisition.

Furthermore, *situational update information*, posted by on-the-ground observers, is actually highly sparse in SM content [C.3]. Therefore, these highly valuable information may be lost if we restrict our system to solely operate on aggregated content, which will be likely dominated by general news and comments, i.e., establish rather coarse-grain, event-level information. On the contrary, aggregation increases the confidence in the reported events. Therefore, we propose an *aggregation-segregation-based* approach, by introducing an additional ***Event Miner*** component: After confirming the general ***Event*** context based on ***Message Aggregation*** ("net-fishing" related tweets) or a feedback-loop from ***Situation Assessment*** or ***Situation Prediction*** [C.4], the ***Event Miner*** specifically seeks to retrieve single tweets which comprise such highly-relevant situational information ("line fishing" specific tweets), such as [T.11], which should be identifiable based on their high information content. These may individually trigger the instantiation of the corresponding ***Crowd Objects*** (e.g., the ***Object*** "DownedTrees" with location Kapoho, which has been only mentioned in a single tweet in our data set, notably [T.11], out of 82 tweets reporting on hurricane damage in Kapoho).

11 Object Integration & Situation Assessment — Merging crowd-sensed events with other information sources.

Finally, ***Crowd Objects*** may be fused with ***Authority Objects***, i.e., data obtained from authority sources, and ultimately serve as input for situation assessment and projection. Fig. 1 shows an example situation that can be derived from the *HawaiiHurricanes* data set, depicting the assessment and evolution of a disaster situation instance capturing and summarizing the chain of events triggered by Hurricane Iselle's landfall on Big Island, which caused major power outages and gradual power restoration due to fallen trees in several communities on the East side of Big Island.

IV. Outlook

In the present work, we devised the architecture of a SAW system for disaster management, which integrates both, *authority* sensors, and *crowd* sensors retrieving disaster-related information from SM. We contributed concepts for enhancing the self-adaptivity of the system and determining event- and situation-level context by highlighting potential feedback loops, which for instance aim at complementing currently partially assessed situations by spawning new crowd-adapters. Whereas our concepts are backed up by initial case studies on real-world Twitter data, further large-scale studies on monitoring different disasters are required to evaluate the applicability towards various types of crisis, determine optimal configurations for the different crowd-sensing and -perception components, and study their potential and limitations.

V. Acknowledgments

This work has been funded by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under grant FFG FIT-IT 829598, FFG BRIDGE 838526, and FFG BRIDGE 832160. We further thank Alexander Porod for implementing quality filtering, and Carina Reiter and Gerald Madlsperger for collecting and analyzing Twitter data.

References

[1] B. Walle and M. Turoff, "Decision support for emergency situations," *Information Systems and e-Business Management*, vol. 6, no. 3, 2008.

[2] A. Sheth, "Citizen Sensing, Social Signals, and Enriching Human Experience," *Internet Computing, IEEE*, vol. 13, no. 4, 2009.

[3] M. A. Cameron *et al.*, "Emergency Situation Awareness from Twitter for Crisis Management," ser. WWW '12 Companion, 2012.

[4] S. Dashti *et al.*, "Supporting Disaster Reconnaissance with Social Media Data: A Design-Oriented Case Study of the 2013 Colorado Floods," in *Proceedings of ISCRAM 2014*, 2014.

[5] J. Dugdale *et al.*, "Social Media and SMS in the Haiti Earthquake," ser. WWW '12 Companion, 2012.

[6] F. Shaw *et al.*, "Sharing news, making sense, saying thanks: Patterns of talk on Twitter during the Queensland floods," *Australian J. of Communication*, vol. 40, no. 1, 2013.

[7] Brian Ulicny, Jakub Moskal, and Mieczyslaw M. Kokar, "Situational Awareness from Social Media," in *Proc. of the Eighth Conf. on Semantic Technologies for Intelligence, Defense, and Security*, 2013.

[8] J. Yin, *et al.*, "Using Social Media to Enhance Emergency Situation Awareness," *IEEE Intelligent Systems*, vol. 27, no. 6, 2012.

[9] H. Purohit and A. Sheth, "Twitris v3: From Citizen Sensing to Analysis, Coordination and Action," in *ICWSM 2013*.

[10] F. Abel *et al.*, "Semantics + Filtering + Search = Twitcident. Exploring Information in Social Web Streams," in *Proc. of the 23rd ACM Conf. on Hypertext and Social Media*, ser. HT '12, 2012.

[11] A. M. MacEachren *et al.*, "SensePlace2: GeoTwitter analytics support for situational awareness," in *IEEE VAST 2011*.

[12] J. Rogstadius *et al.*, "CrisisTracker: Crowdsourced Social Media Curation for Disaster Awareness," *IBM J. of Research and Development*, 2013.

[13] S. Kumar *et al.*, "TweetTracker: An Analysis Tool for Humanitarian and Disaster Relief," in *Fifth Intl. AAAI Conf. on Weblogs and Social Media*, 2011.

[14] T. Sakaki *et al.*, "Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors," ser. WWW '10, 2010.

[15] H. Smid *et al.*, "Canary in a Coal Mine: Monitoring Air Quality and Detecting Environmental Incidents by Harvesting Twitter," in *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, 2011.

[16] G. Jakobson *et al.*, "Situation-Aware Multi-Agent System for Disaster Relief Operations Management," in *ISCRAM 2006*.

[17] J. Llinas *et al.*, "Revisiting the JDL Data Fusion Model II," in *FUSION 2004*.

[18] A. Salfinger, S. Girtelschmid, B. Pröll, W. Retschitzegger, and W. Schwinger, "Crowd-Sensing Meets Situation Awareness - A Research Roadmap for Crisis Management," in *HICSS-48*, 2015.

[19] S. Vieweg *et al.*, "Microblogging During Two Natural Hazards Events: What Twitter May Contribute to Situational Awareness," in *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, 2010.

[20] B. Pröll, W. Retschitzegger, W. Schwinger *et al.*, "crowdSA - Crowdsourced Situation Awareness for Crisis Management," in *Proc. of SMERST 2013*.

[21] N. Baumgartner, W. Gottesheim, S. Mitsch, W. Retschitzegger, and W. Schwinger, "BeAware!—Situation awareness, the ontology-driven way," *Intl. J. of Data and Knowledge Engineering*, vol. 69, 2010.

[22] N. Baumgartner, S. Mitsch, A. Müller, W. Retschitzegger, A. Salfinger, and W. Schwinger, "A Tour of BeAware! – A situation awareness framework for control centers," *Information Fusion*, vol. 20, 2014.

[23] C. Matheus *et al.*, "SAWA: An Assistant for Higher-Level Fusion and Situation Awareness," in *Proc. of SPIE Conference on Multisensor, Multisource Information Fusion*, 2005.

[24] A. Salfinger, W. Retschitzegger, and W. Schwinger, "Staying Aware in an Evolving World — Specifying and Tracking Evolving Situations," in *CogSIMA 2014*.

[25] P. Kogut *et al.*, "UML for Ontology Development," *Knowl. Eng. Rev.*, vol. 17, no. 1, pp. 61–64, 2002.

[26] C. Matheus *et al.*, "A core ontology for situation awareness," in *Proc. of the Sixth Intl. Conf. of Information Fusion*, vol. 1, 2003.

[27] M. Kokar *et al.*, "Situation tracking: The Concept and a Scenario," in *IEEE Military Communications Conf., 2008*.

[28] A. Salfinger, D. Neidhart, W. Retschitzegger, W. Schwinger, and S. Mitsch, "SEM^2 Suite — Towards a Tool Suite for Supporting Knowledge Management in Situation Awareness Systems," in *IRI 2014*.

[29] M. Cunningham *et al.*, "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications," in *Proc. of the 40th Anniversary Meeting of the Assoc. for Computational Linguistics (ACL'02)*, 2002.

[30] K. Bontcheva *et al.*, "TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text," in *Proc. of the Intl. Conf. on Recent Advances in Natural Language Processing*, 2013.

[31] H. Purohit *et al.*, "Identifying Seekers and Suppliers in Social Media Communities to Support Crisis Coordination," *Computer Supported Cooperative Work (CSCW)*, vol. 23, no. 4-6, 2014.

[32] D. d. Wrachien *et al.*, "Ontology for flood management: a proposal," *Flood Recovery, Innovation and Response III*, vol. 159, p. 3, 2012.

[33] L. Hong *et al.*, "Language Matters In Twitter: A Large Scale Study," in *Fifth Intl. AAAI Conference on Weblogs and Social Media*, 2011.

[34] Y. Ikawa *et al.*, "Location-based Insights from the Social Web," ser. WWW '13 Companion, 2013.

[35] J. L. Leidner, "Toponym resolution in text," Ph.D. dissertation, 2007.

[36] M. Imran *et al.*, "AIDR: Artificial Intelligence for Disaster Response," ser. WWW Companion '14, 2014.

[37] M. Nagarajan *et al.*, "Spatio-Temporal-Thematic Analysis of Citizen Sensor Data: Challenges and Experiences," in *Proc. of the 10th Intl. Conf. on Web Information Systems Engineering*, ser. WISE '09, 2009.