Situation Recognition: An Evolving Problem for Heterogeneous Dynamic Big Multimedia Data

Vivek K. Singh, Mingyan Gao, and Ramesh Jain,

University of California, Irvine

Abstract

With the growth in social media, internet of things, and planetary-scale sensing there is an unprecedented need to assimilate spatio-temporally distributed multimedia streams into actionable information. Consequently the concepts like objects, scenes, and events, need to be extended to recognize ***situations*** (e.g. epidemics, traffic jams, seasons, flash mobs). This paper motivates and computationally grounds the problem of situation recognition. It describes a systematic approach for combining multimodal real-time big data into actionable situations. Specifically it presents a generic approach for *modeling* and *recognizing situations*. A set of generic building blocks and guidelines help the domain experts model their situations of interest. The created models can be tested, refined, and deployed into practice using a developed system (EventShop). Results of applying this approach to create multiple situation-aware applications by combining heterogeneous streams (e.g. Twitter, Google Insights, Satellite imagery, Census) are presented.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**], D.3.3 [**Information Systems**]: World Wide Web – *Social Networks*

Keywords

Events, Situation, Situation Detection, Modeling, Situation Awareness, Social Networks, Sensor Networks

1. INTRODUCTION

We are living in an age of abundance [1]. Humanity is more connected than ever before. With the growth trends in *social media*, *multimodal mobile sensing*, and *location driven sensing*, increasingly larger parts of human life are getting digitized and becoming available in the Cloud for sense making.

The fundamental problem of sense-making is that of *making sense of the observed data*. For multiple decades, researchers have been building approaches like entity resolution, object detection, and scene recognition, to understand different aspects of the observed world. Unlike the past though, now we do not need to undertake sense-making based on data coming from a single media element, modality, time-frame, or location of media capture. Real world phenomena are now being observed by multiple media streams, each complementing the other in terms of data characteristics, observed features, perspectives, and vantage points. Each of these multimedia streams can now be assumed to be available in real-time and increasingly larger portion of these come inscribed with space and time semantics. The number of such media elements available (e.g. Tweets, Flickr posts, sensors updates) is already in the order of trillions [20], and computing resources required for analyzing them are easily available. We expect this trend to continue; and mobile devices to become the biggest *producers* and *consumers* of multimedia data. Hence sense-making from real-time multimodal location-aware data will be the 'holy-grail' of Computer Science problems for the coming decade.

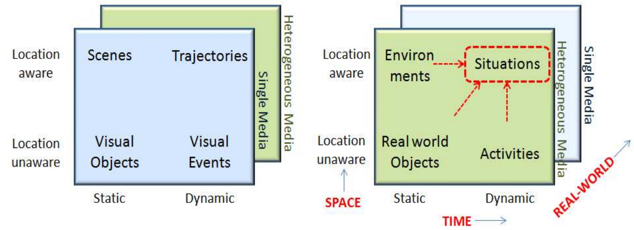

Figure 1: Different Types of Concepts can be detected in different data availability settings. Single media, such as images, results in concepts more in images than in the real world, but using different media it is possible to detect concepts in the real world

The time is right now to channel the lessons learnt from detecting ***intra-media concepts*** (i.e. those which manifest themselves, and can be detected within a single media object e.g. a tree, or a chair *in* an image), to define and making quick progress on detecting ***evolving concepts*** (i.e. those which occur in real world, are constantly evolving, and inherently manifest themselves over heterogeneous multimedia streams from numerous sources). As a simple example, we may now look beyond the problem of creating a *tree* detector and/or testing it over Millions of Flickr images; to that of using a stream of Billions of such images and other available data to detect seasonal patterns, plant disease spreads, deforestation trends, or global warming. These are the problems which could not be tackled earlier because of the lack of data and computational resources; but those are no longer the bottlenecks.

As shown in Figure 1, ***Situation recognition*** builds on and extends object recognition, scene recognition, activity and event recognition, and complex event processing. Examples of relevant situations are all around us including beautiful-days/ hurricanes/ wildfires, traffic (jams / smooth/ normal), economic recessions/ booms, block-busters, droughts/ great-monsoons, seasons (early-fall/ fall/ late-fall), demonstrations/ celebrations, social uprisings/ happiness-index, flash-mobs, flocking and so on. They vary across, and affect all aspects of human lives – health, natural disaster, traffic, economy, social reforms, and business decisions. Detecting situations in time to take appropriate actions for saving lives and resources can transform multiple aspects of human life.

The challenges in Situation recognition would be fundamentally different from those in object or event recognition. In effect, this problem brings us back to the drawing boards as we establish the process of concept detection from very heterogeneous, very unstructured, real time, big data. For example, we can no longer just ***accept*** heterogeneity, or ***allow*** multiple data streams; we need to ***expect*** them and ***capitalize*** on them. We need to focus on recognition of real world phenomena based on their footprints across multiple heterogeneous media. This will allow us to solve practical human problems by correlating data ranging from social media, to sensor networks, and satellite data. Consider hurricane mitigation as an example. The data streams for hurricane status (e.g. NOAA.gov), weather forecast (weather.com), population demographics (census.gov), rescue shelters (redcross.org), and traffic directions (maps.google.com) are all freely available on the Web. But we still lack the computational frameworks to unify and process such heterogeneous streams to solve practical problems.

Multimedia research community is particularly well positioned to take on the challenge of Situation Recognition. First, the community's core competence lies in handling heterogeneous media (audio, video, text, phone logs, micro-blogs, sensors etc.). Equally importantly, it is the only research community which has studied concept detection across both time (event detection), and space (like spatial organization of pixels in images). The tools for raster image processing translate directly onto spatial data layouts and notions of neighborhood, regions, boundaries, and motion vectors translate seamlessly across to spatio-temporal analysis. We just need to change the perspective and focus on these notions in the real world rather than the media silos.

Building the tools and techniques to handle the various challenges in defining new concepts, dealing with big data volume, and building cross-media processing tools require a long term community effort. This paper describes the first systematic attempt towards detecting situations in the context of large scale spatio-temporal multimedia streams. Correspondingly, the scope of this paper is to identify and illustrate a ***generic approach for modeling and recognizing situations***. Specifically, we identify 3 main problems essential for situation recognition:

1) The concept of 'situation' is still ill defined; previous attempts ignored the role of big, real-time, heterogeneous, spatio-temporal streams.

2) There is a lack of conceptual tools to help users model their situations of interest, and

3) No tools are available to rapidly implement these situation models, and reevaluate and refine as required.

This paper describes an approach for tackling each of these challenges. After surveying different interpretations of situation, a computational definition for it is developed. Next, we describe a step by step approach to help domain experts in creating computational recognition models for different situations. We provide guidelines and support a design process to make sure that the models generated are explicit, actionable, and computable. A toolkit to rapidly implement and evaluate these situation recognition models helps building solutions. The modeling tool and the implementation are based on the use of space and time as the unifying axes for heterogeneous data, and spatio-temporal features as those important to application designers to differentiate between classes of spatio-temporal phenomena. The implementation of the recognition models is built upon transposing the spatio-temporal operators into raster image and video operations under the hood. Once refined until satisfactory levels of detection/ recognition are achieved, the detectors can be used to build situation based applications to generate appropriate information and personalized actions.

We will discuss the use of the proposed approach across multiple applications dealing with diverse data streams.

2. SITUATION: DEFINITION

There has been a large amount of work done in the areas like ubiquitous/pervasive computing [9], context-aware computing [31], mobile application software, aviation/air traffic control [4, 6, 12], robotics, industrial control [11], military command and control [5], surveillance [22], linguistics [10], stock market databases [30], and multimedia analysis[23] on situation modeling, situation awareness, situation calculus, situation control, and situation semantics. The interpretation of *situation* however is different across different areas and even across different works within the same area.

To highlight the common themes and illustrate the diversity in interpretations, we present some sample definitions here.

- *Endsley, 1988: "the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future"*
- *Merriam-Webster dictionary: "relative position or combination of circumstances at a certain moment"*
- *McCarthy, 1969: "A situation is a finite sequence of actions."*
- *Yau, 2006: "A situation is a set of contexts in the application over a period of time that affects future system behavior"*
- *Dietrich, 2003: "…extensive information about the environment to be collected from all sensors independent of their interface technology. Data is transformed into abstract symbols. A combination of symbols leads to representation of current situations…which can be detected"*

Some common traits as well as the dissimilarities amongst different definitions are clear. Most telling perhaps is the observation by Jakobson et al that "…being a relatively new field, there is a clear lack of theoretic well-grounded common definitions, which may be useful across different domains." [18].

We decided to focus on the commonalities across definitions, and identified the following notions to reverberate across definitions:

1) **Goal based (GB):** Situations need to be defined for an application or a purpose.
2) **Space and time (ST):** Situations capture and represent a volume of space and/or time.
3) **Future actions (FA):** Situations can be used for future prediction and/or action taking.
4) **Abstraction (AB):** Situations signify some form of perception, or symbolic representation for higher understanding.

Further while some definitions were **computationally grounded (CG)** in data (e.g. Endsley, Dietirch), others were abstract (e.g. Barwise, Merriam-Webster). Here, we summarize some of the definitions surveyed based on these axes:

Work	Goal Based	Space Time	Future Actions	Abstr action	Computat ionally Grounded
[4] Endsley, 1988		X	X	X	X
[5] Moray, 2004		o		X	
[6] Adam, 1993	X		X		
[7] Jeannot, 2003	X				
[8] McCarthy, 1969			X		
[9] Yau, 2006	X		X		X
[10] Barwise, 1971		X		X	
[11] Dietrich, 2004				X	X
[12] Sarter, 1991		o		X	
[13]Dominguez,1994	X		X	X	X
[14] Smith, 1995	X	o	X	X	
[15] Dostal, 2007		o		X	
Merriam-Webster		o			
[16] Singh, 2009	X		X		X
[21] Steinberg, 1999	X		X	X	o

Note: '**o**' indicates partial support.

2.1 Proposed definition

Based on observing the common traits as well as a focus on staying computationally grounded, we define a situation as:

"An actionable abstraction of observed spatio-temporal descriptors"

Going right to left, let us consider each of the terms used in this definition:

a) descriptors: We adopt the approach of quantifying an abstract/inexact notion based on its observed characteristics. This underlines that we want to computationally ground the definition.

b) spatio-temporal: This work's focus, scope, as well as the most common connotation associated with 'situations', is on spatio-temporal data.

c) observed: We focus on the 'observable' part of the world. Meta-physical as well as physical aspects which cannot be measured by sensors present are simply outside the scope of problems we can tackle.

d) abstraction: We want to create information at a much higher level than sensor measurements or even their lower level derivations. Decision makers typically focus on higher (knowledge) level abstractions while ignoring the lower level details.

e) actionable: The top level descriptors and abstractions need to be chosen based on the application domain, and the associated output state-space. Hence our focus is on creating a representation (e.g. classification) which maps the lower level details into one concrete output decision descriptor. Hence, we are not interested in *any* higher level abstraction, but rather the *specific* one which supports decision making in the application considered.

As can be noticed, this definition operationalizes the reverberating threads found across different definitions in literature, and computationally grounds them.

The difference in the definition of 'situation' as a concept also exemplifies the difference between our approach at tackling Situation recognition and similar efforts in Context-aware computing, Social media mining, Geographical Information Systems (GIS), Active databases, Multimedia analysis, and Complex event processing literature. For example, GIS community has studied spatial data analysis extensively but paid lesser attention to temporal aspects or real time streams. Complex Event processing research on the other hand focuses on real-time stream analysis but rarely considers the spatial semantics. In effect we are building upon and extending these efforts to make progress towards the problem of situation detection.

3. OVERALL FRAMEWORK

3.1 Overview

We ground our discussion on modeling and detecting situations onto a generic framework which this work builds towards. This framework focuses on combining heterogeneous real-time data streams into actionable situations. The framework focuses on the spatio-temporal commonality across streams to integrate them. By using a simple unified representation (based on space-time-theme), it indexes and organizes all data into a common representation. Similarly, for going from individual data nuggets (micro-events) to macro-situations it uses a set of generic spatio-temporal analysis operators. A basic assumption in this approach is that spatio-temporal situations are determined by evaluating a large number of data streams that represent different attributes measured by either physical sensors or observed by human-sensors.

The fundamental data structure being operated on in this framework for combining spatio-temporal data is E-mage, each cell of which captures a value associated with a particular theme at a particular spatio-temporal coordinate. The use of grid is based on the understanding that grids are the fundamental data structure used by humans to understand and analyze spatial data (e.g. maps, satellite images). They also capture the semantics and notion of spatial neighborhood very elegantly, and geographical-joins [19] between data streams reduce to simple overlaying of grids. An example of an E-mage is shown in Figure 2. E-mages were first defined in [2].

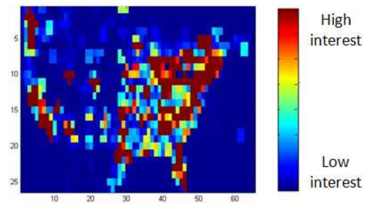

Figure 2: An E-mage showing user interest across mainland US in terms of number of tweets containing the term 'iphone' on 11th Jun 2009

3.2 Detecting Situations from Heterogeneous Streams

The process of moving from heterogeneous streams to situations is shown in Figure 3. The unified STT format employed (level 1) records the data originating from any spatio-temporal bounding box using its numeric value. Aggregating such data results in two dimensional data grids (level 2). At each level the data can also be characterized for analytics. The situational descriptor (level 3) is defined by the user (application expert) as a function of different spatio-temporal characteristics.

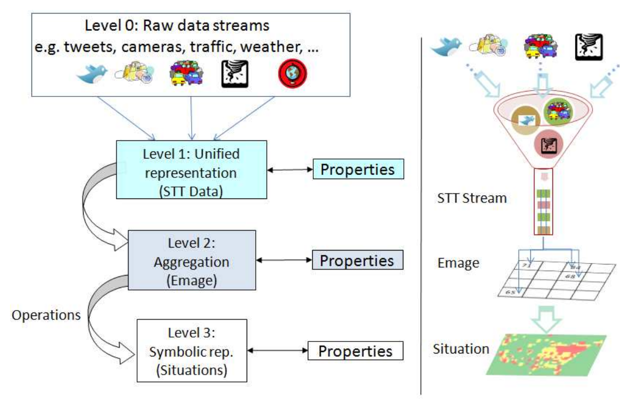

Figure 3: Approach for detecting Situations

3.3 Data Representation Levels

Level 0: Diverse Raw Data

The framework supports data from different sources. Any sensor data can be associated to a stream based on its location and frequency of creation. Human sensor data, such as tweets and status updates can also be analyzed and converted to measurements related to a particular theme or attribute. Some data sources have tables or databases that are frequently updated to give certain sensory data collected by different agencies. Hence, we support as many different types of raw data as may be relevant. The types of data streams supported in the system will evolve as it is used for diverse applications. For computational purposes we normalize all data streams to numeric streams.

Level 1: Unified Representation

Heterogeneous data needs to be unified. Also, too much data can lead to high cognitive and data processing costs. This layer converts individual attributes into information in terms of 'what-when-where' i.e. STTPoint, and facilitates aggregation of information in next (i.e. E-mage) level.

Level 2: Aggregation

Spatial data can be naturally represented in the form of spatial grids with thematic attributes. As explained, the framework considers E-mages, and E-mage Streams as its data model. This image-like representation allows application of a rich collection of image and video processing operators ranging from segmentation, aggregation, detecting spatial and temporal patterns, and tracking patterns across space and time. Such a representation also aids easy visualization, and provides an intuitive query and mental model.

Level 3: Situation Detection and Representation

The situation at a location is characterized based on spatio-temporal descriptors determined by using appropriate operators at level 2. The final step in situation detection is a classification operation that uses domain knowledge to assign appropriate class to each cell. This classification results in a segmentation of an E-mage into areas characterized by the situation there. Once we know the situation, appropriate actions can be taken.

3.4 Operators

Multiple operators need to be provided for analysis and characterization of temporal E-mage streams. The operators considered include Filter, Aggregation, Classification, Spatio-temporal Characterization, and Spatio-temporal Pattern matching.

3.5 Personalized Action Alerts

The situations detected can be combined with individual user parameters for customized action taking. We focus on action recommendation using the E-C-A (Event-Condition-Action) [25] approach. The individual parameters can be spatio-temporal coordinates, as well as personal micro-events (e.g. 'sneezing') detected. The spatio-temporal coordinates can be used to direct users to nearest location satisfying certain conditions. Multiple such E-C-A templates can be registered to provide customized alerts to all recipients.

4. SITUATION MODELING

Situation modeling is the process of conceptually defining what constitutes an actionable situation in the application designer's domain. It allows the designer to externalize what she means by a specific situation of interest (e.g. an 'Epidemic'). Building this model in terms of conceptual building blocks rather than directly implementing them in code has multiple advantages. First, the application designers get to focus on the Big-Picture rather than getting bogged down by the implementation details. Next, such a process encourages a goal-driven thinking rather than an availability driven thinking [27]. Lastly, the modeling in terms of generic blocks allows for easy reuse of components across applications.

To aid the creation of different situation models, we provide:

1) *The 'building blocks'*
    a. *Operators*
    b. *Operands*
2) *An prescriptive approach for modeling situations using the operators and operands*
3) *Steps and guidelines for refining the models so that they are computable and explicit.*

4.1 Operators and Operands

We want to provide constructs which are generic enough to capture most of the common requirements across different applications. At the same time these constructs need to be well-defined and quantifiable. Considering these two (often competing) considerations we have defined the following set of operands and operators.

4.1.1 Operands

The operands in the framework are conceptual features, data representation levels, and supporting meta-data on which different operators may be applied.

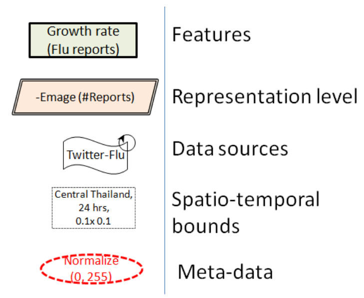

Figure 4: Operands for Situation Modeling

The ***operands*** defined (also see Figure 4) are:

1) Feature: Any spatio-temporal descriptor that contributes towards defining the overall situation. (e.g. growth rate of Flu reports)

2) Representation level: Data representation levels required (e.g. STT nuggets, Emages)

3) Data source: The resource for obtaining data in any supported format.

4) **ST bounds:** The Spatio-temporal bounding boxes to consider when obtaining data-streams or evaluating any features.

5) Meta-data: Any additional details (e.g. Operator types, normalization bounds, output variables, thresholds) required for complete specification of the features or operators.

4.1.2 Operators

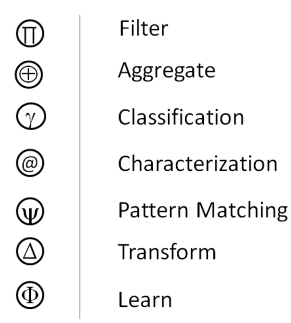

Figure 5: Operators for Situation Modeling

The ***operators*** defined (refer Figure 5) are:

1) Filter (∏): This allows for selection of data based on space, time, theme, or value parameters.

2) Aggregation (⊕): This allows for features to be combined based on mathematical operators.

3) Classification (γ): This operator classifies the values into different segments representing different semantic entities.

4) Characterization [spatio-temporal] (ϕ): This operator handles derivation of different spatio-temporally relevant attributes (e.g., epicenter, density, shape) for any data stream.

5) Pattern Matching [spatio-temporal] (ψ): This operator allows users to study how closely the captured phenomena match known patterns or related historical data.

6) Transform (Δ): This allows the data at any layer to be transformed into the next (higher) layer. It can be for:

> ***a) Data source to STT (Level 0 to Level 1):*** The wrappers to translate heterogeneous data into STT data nuggets.
>
> ***b) STT to Emages (Level 1 to Level 2):*** Combining STT nuggets into an aggregated Emage representation

7) Learn (Φ): Sometimes the precise weight of the identified features on the situational descriptor might be unknown to the expert. In such cases the expert is required to identify the learning data source from which the system can automatically infer such values (e.g. using Machine Learning).

4.2 The Wizard for modeling situations

We also provide a prescriptive approach for creating situation models.

As shown in Listing 1 the first step requires the application designer to identify the output state space (i.e. *range* for the output descriptor). Next she needs to identify the spatio-temporal bounds being considered. Next step is identifying the relevant features useful in defining the situation output. If it is an imprecise classification type of problem, then the expert is required to identify the data source for 'learning' how the different features identified affect the situation classification.

/* Input: Situation descriptor (possibly complex and vague) */
/* Output: List of intermediate descriptors, data sources, and representation levels required */
Get_components (v){
1) Identify output state space
  - Numeric or Classes
2) Identify S-T bounds
3) Identify component features; $v = f(v_1, \ldots, v_k)$
  - If (type = imprecise)
    - identify learning data source, method

4) ForEach (feature v_i){
  - If (atomic)
    - Identify Data source.
      - Type, URL, ST bounds
    - Identify highest Rep. level reqd.
    - Identify operations.
  - Else
    - Get_components(v_i)

}

Listing 1: The approach for generating conceptual situation models

For each of the features identified above, user needs to identify the data sources, representation levels required and the variables/themes needed for the transformation across levels. If the feature considered is atomic (i.e. detectable using well defined operations on a data stream) it is added to the model. In case the feature is not well defined yet, a recursive call is invoked onto the same algorithm to identify the relevant components and details for the one-lower level feature.

As shown in Figure 6, such a process repeats itself in a recursive manner (akin to depth first search) until the high level situation description has been partitioned into explicitly observable or computable components.

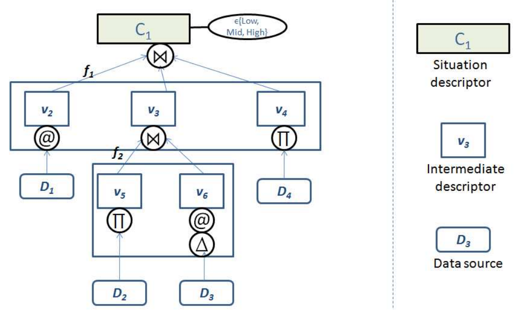

Figure 6: Recursive approach for defining situation variables

4.3 Enhancing the model

4.3.1 Refining the model

A model created by the process above captures a baseline representation of an application designer's model of a situation of interest. Just like E/R modeling [24] this process may require multiple iterations before the experts agree on a 'good' model. Suggested criteria to accept a 'good' model are:

1. *Do Features identified provide enough discriminative power to the model?*
2. *Does the data stream chosen capture the semantics of the feature selected?*
3. *Are there any cyclic reasoning or cyclic dependencies in the features selected?*

4.3.2 Instantiating the model

The process of instantiation involves adding all the relevant details to make the operators computably explicit i.e. contain enough detail to be translated into code if required. This requires the following steps:

1. *Provide necessary parameters for operators in the model.*
2. *Refine, if necessary*

The model created after undertaking all these steps would capture all the details (e.g. Operator types, normalization bounds, thresholds) required for implementation. The full details of all the parameters required to quantify each operator are discussed in [3]. Note that any platform specific details (e.g. implementation language, language related issues, data types, memory management) are still not (and not supposed to be) part of this model.

Once satisfied with the model, the application designers can translate the model into code using a set of libraries which can be called programmatically or using a graphical tool like EventShop. With some training, and advancement in UI, we might see the domain experts themselves doing this translation, but we leave that outside the scope of our current discussion.

4.4 Example: Modeling *Epidemic Outbreaks*

Let us illustrate the process of situation modeling by considering 'epidemic outbreaks'. Given as-is, 'epidemic outbreak' is a vague undefined notion. In fact not even all experts agree on what constitutes an Epidemic. Here we discuss the workflow for one possible modeling of epidemics.

Following section 4.2, we first identify the output state space (i.e. required *classification into low, mid, and high* risk of outbreak). As shown in Figure 7 we identify the spatio-temporal bounds being considered <*USA, with a spatial resolution of 0.1 latitude X 0.1 longitude, and re-evaluation to be made every 5 minutes*>. Next step is identifying the relevant features useful in defining the situation output. We define 'epidemic outbreaks' as a classification on 'growing unusual activity'. While this is a single feature, it is not atomic (i.e. cannot be derived directly using one data source). Hence we follow the process recursively, and try to model 'growing unusual activity'. This feature is defined based on two component features: 'Unusual activity' and 'Growth Rate'. It turns out that 'Unusual activity' is also not atomic, and needs to be split into the features of 'historical activity level' and 'growth rate'. Let's assume that the historical activity level is available from a curated database and current activity level can be measured based on the frequency of terms indicating ILI (Influenza-Like-Illness) on Twitter stream. Similarly, the growth rate can be measured from Twitter stream.

Hence, now we have three ('leaf node') features which can each be defined using a single data source and hence the modeling is complete. In effect we have split a vague concept (epidemic) into features such that each of them can be derived from a known data source.

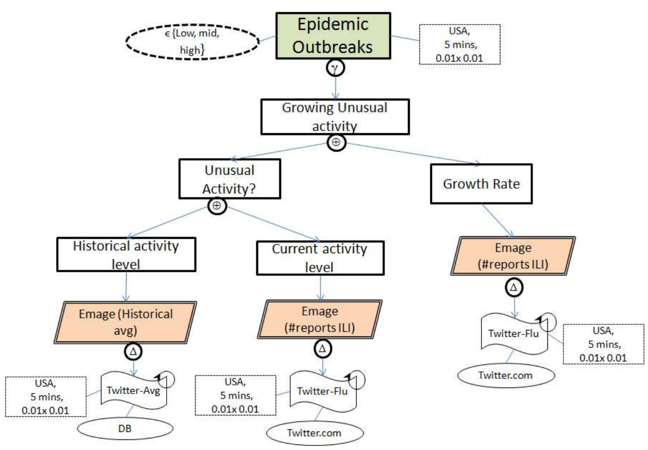

Figure 7: Base model created for epidemic outbreaks

In practice, application designers may not be satisfied with the first created model. For example, let's consider 'current activity level' which has been defined based on the number of Influenza-Like-Illness (ILI) reports observed from each location. It may be better to regularize this value based on the population at each location. This leads to changes in part of the model shown in Figure 8.

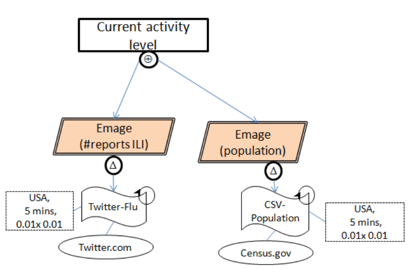

Figure 8: Situation model: Changes made in refinement phase

Last phase in model creation is that of 'instantiating' it. This step involves adding the relevant details about the exact operation to be performed and the associated parameters. Further, certain parameters/ data sources may need to be refined. Here we scale the population Emage to the range [0,100] to be comparable to the incidents reported.

Figure 9 shows the model with the relevant details added (e.g. [30, 70] as the thresholds for classification; 'And' as the precise operation used for aggregation). Once this step is complete, the created model can be evaluated using EventShop or a similar validation toolkit.

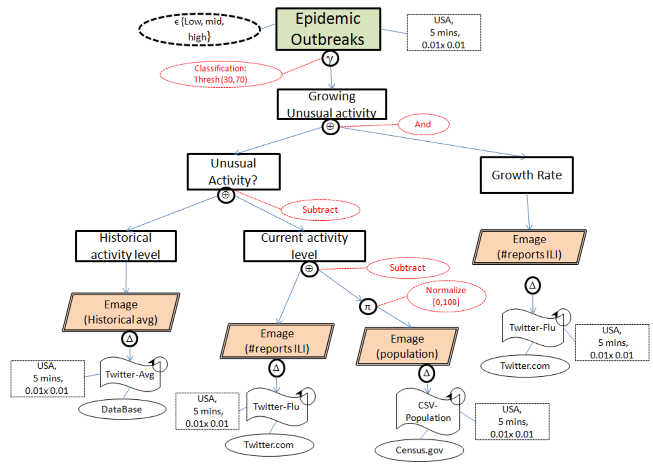

Figure 9: Situation model after the instantiation phase (details added in Red)

5. EVENTSHOP: A TOOLKIT FOR RAPID VALIDATION OF SITUATION MODELS

To easily validate, refine, and re-evaluate the situation models we have built a system called EventShop. EventShop includes a front end GUI (Graphical User Interface) and a back end stream processing engine. In the front end, EventShop borrows the idea of PhotoShop by providing a user-friendly GUI that allows end users to select different streams and configure situation filters. Users are also provided with a GUI tool which allows them to send personalized alerts to relevant people. Based on the information of registered data sources, EventShop continuously ingests spatio-temporal-thematic data streams and converts them to E-mage streams. Meantime, directed by the registered queries, EventShop pulls E-mage streams from data ingestors to query processor, which process the E-mage streams in each of the instantiated query operators. Besides being converted to E-mage streams, the raw data stream, (e.g. tweet stream) is also made persistent into raw data storage. Raw data together with query results provides necessary personal as well as local situation information to Personalized Alert Unit which can be used for creating situation aware applications.

A snapshot of EventShop is shown in Figure 10. The basic components are:

a) Data-source Panel: To register different data sources into the system.

b) Operators Panel: Different operators that can be applied to any of the data sources.

c) Intermediate Query Panel: A textual representation of the intermediate query currently being composed by the user.

d) Registered Queries: A list of configured queries registered with the system.

e) Results Panel: To see the output of the query (which can be presented on a map, timeline, as a numeric value or a combination).

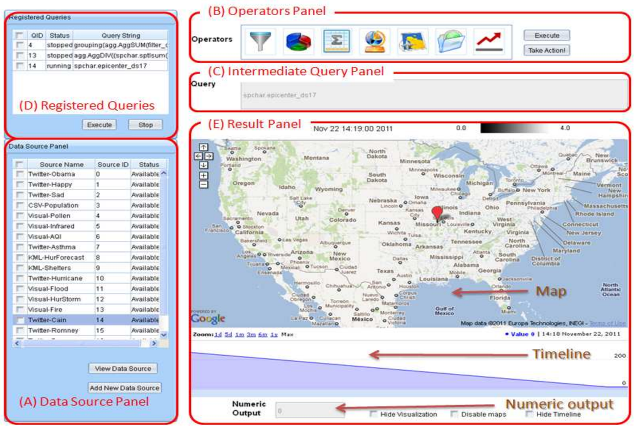

Figure 10. A snapshot of EventShop

Implementation of runtime operators makes use of OpenCV package. More details on EventShop are presented in [3], and details of translating spatio-temporal operators into image/video processing operators are similar to [2]. While the primary goal of EventShop is to work on real-time streams it can also be configured to deal with archives of data streams entering the system at a configured rate.

Note that the implementation in EventShop also provided us with evidence on the explicit and computable nature of each of the operators defined. Also note that we have decided to keep the learning operator outside the scope of the first implementation of EventShop.

The system is available at [28]. We plan to release the system to the open source community in near future.

6. EXPERIMENTAL VALIDATION

We have tested our approach for situation modeling as well as detection across multiple applications. Our tested scenarios include Hurricane detection and mitigation, identifying weather patterns (e.g. Fall colors in New England), Influence patterns for different political figures, Identifying demand hot-spots of business products, Allergy risk and recommendation, Flu outbreak, Wildfires, Global Warming Index, Quality of Living, and Flood mitigation. We select three of these applications for our discussion here. We use them as representative examples to give a view of the diverse applications which can be studied using this approach.

First example application is that of Epidemic Outbreaks. This example extends the discussion in section 4.4 and completes the lifecycle of a concept, from situation modeling to its actual implementation using EventShop. Second application is that of detecting large scale wild fires in California. Working on archived satellite and social streams, this application allows us to compare the performance of the created situation recognition models with the ground truth data. The third case study presents the results of applying this approach to aid real end users during the recent floods in Thailand. This example illustrates the life-cycle of going from situation models, to building situation-aware applications which can potentially help millions of users in real world situations.

6.1 Detecting *Epidemic Outbreaks*

To validate if the models created following the approach described in section 4 are indeed explicit and computable, we took the model and configured it in EventShop. A video capture of the process is available at [29]. In fact it turned out to be a very straightforward exercise as all the necessary building blocks and their configuration parameters were already specified in the model. The ability to translate the models into the system was akin to translating a schema into database tables once all the fields have been identified.

For the current purpose we used an average of tweets on ILI for the last month to be the historical average level. We let the system run for two weeks (Apr/30/12-May/13/12) with real-time Twitter data feeds passing through the created filter, and (thankfully!) saw no severe Epidemic outbreak risks. Sample Emages for the different data streams are shown in Figure 11 and a sample of the configured detector's result is shown in Figure 12.

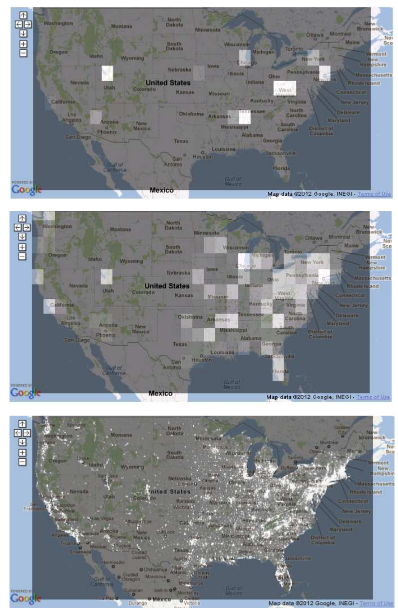

Figure 11: Emage for (a) Reports on Flu (brighter indicates more reports), (b) Historical average, (c) Population

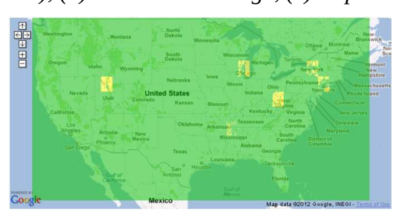

Figure 12: 'Epidemic outbreak' risk level

6.2 Detecting *Wildfires*

Wildfires affect large portions of human ecology and often last days and weeks while spreading over large spatial boundaries. It was estimated that tropical fires around the world destroyed about 15 X 10^6 km^2 of forests in the last decade [26]. Quantitative information about the spatial and temporal distribution of fires is important for forest protection and in the management of forest resources. It is also indispensable to such disciplines as ecology, wildlife management and atmospheric chemistry.

Hence we decided to build a computational model for recognizing wildfires. For this we approached a domain expert (a research scientist in Earth Science department at our university) and requested her to volunteer for our case study.

Based on the process described in section 4.2 and her expertise in the area of satellite based fire detection, she created a situation model shown in Figure 13. This model is loosely based on the algorithm described in [17]. It focuses on using satellite data to detect large wildfires. Specifically it focuses on anomalies in inter-band variation between 4µm and 11µm wavelength radiations to detect wildfires. This inter-band variation can only be observed in unclouded regions; which are identified by analyzing the 12µm band's radiation levels.

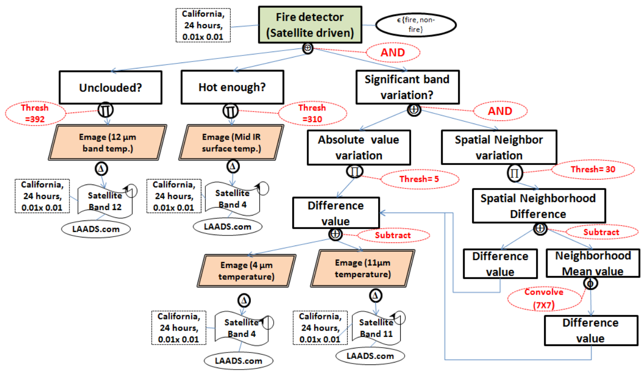

Figure 13: 'Wildfire' recognition model using satellite data

We configured this model into EventShop and detected the various wildfire situations across California based on archives of satellite data streams. The archive of satellite data was obtained from NASA's LAADS website. Using this model we were able to achieve ~74% precision (refer Figure 17) at detecting large fires (>1000 m^2) over last 2 years in California. The ground truth used for comparison was obtained from the website of the California Department of Forestry and Fire Protection.

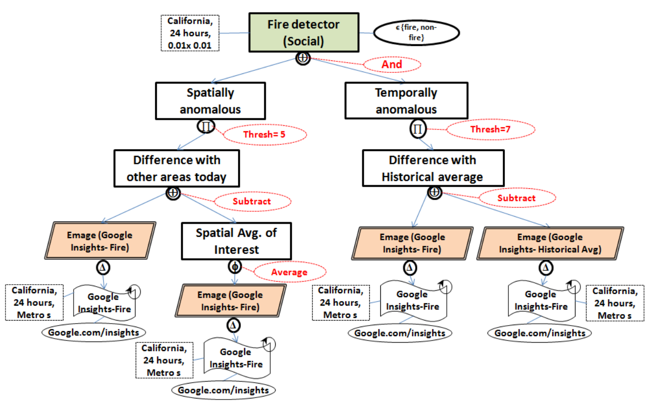

Figure 14: 'Wildfire' recognition model using social data

We discussed with the expert and built another model (see Figure 14) using purely the social media (Fire related search queries made on Google from each location) data, and configured it onto EventShop. Note that the spatial granularity now is much coarser (data is available at 'metro area' level), but it complements the satellite based detection especially in cases where fire occurred in clouded regions, or was brief but affected large human populations.

This model could detect Fire situations in the correct time-frame with ~70% accuracy. We consider this by itself to be an interesting finding that indicates that spatio-temporal nuggets (millions of search query logs) can be combined to create the same effective information as earlier limited to satellites or the proverbial 'God's view'[20].

Lastly we decided to combine the two detection approaches and create a unified situation detector which simply combines the two detectors (see Figure 15). The combined detector could detect more than 90% of the large fires in California. A sample output is shown in Figure 16 and a video capture of the filter configuration and results observed is available at [29].

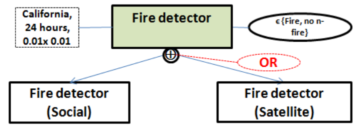

Figure 15: 'Wildfire' recognition model using satellite + social data

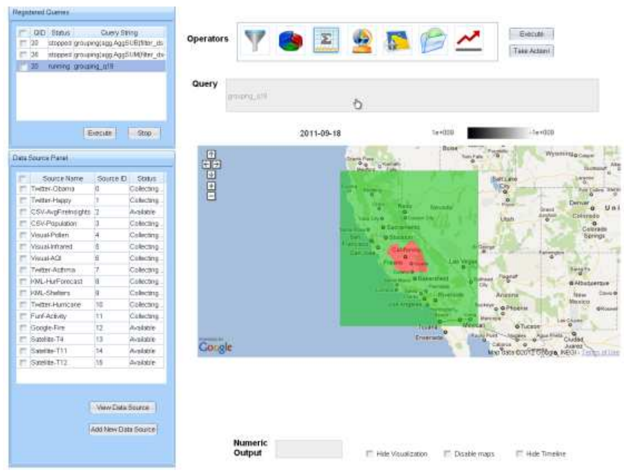

Figure 16: 'Wildfire' recognition model using satellite data

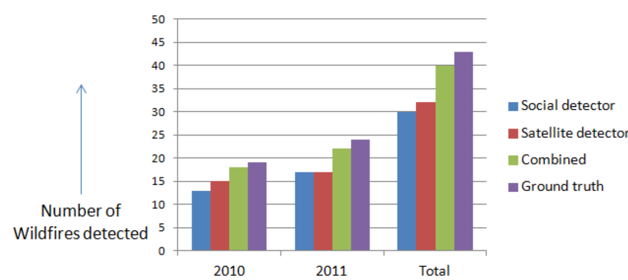

Figure 17: Recognition performance at detecting 'Wildfires' across California over last 2 years

6.3 Building situation aware applications: Thailand Flood Risk Recommendations

After verifying the veracity of the approach at detecting spatio-temporal situations from archives of data streams, let's consider its application at processing real-time data streams to detect evolving situations to help people.

We applied EventShop for the goal of suggesting safe locations to people who were trapped in risky situations in Thailand flood. The situation model for defining risk levels is shown in Figure 18 (a graduate student from Thailand whose family was affected by the floods acted as our application expert in this study).

The idea is to segment flooding areas into three groups based on flooding condition and shelter sufficiency. The model combines the information about the water depth level with the nearby shelter availability to identify areas which have high levels of water but no open shelters. The shelter coverage has been defined based on a Gaussian coverage assumed for each shelter location. The data on the water level was made available by Google.org and was updated roughly every 6 hours. The locations of currently active shelter locations) were constantly being updated by the at-the-ground volunteers in Thailand. We could use this source as a data stream from http://shelter.thaiflood.com/webservice/request.kml .

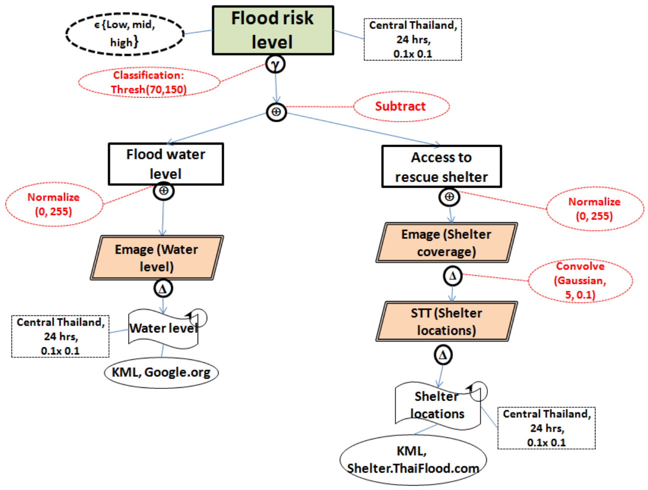

Figure 18: 'Flood threat level' detection model

Figure 19: Sample Emages showing (a) Flood Water Level (brighter implies higher water level), and (b) Shelter Coverage

We ran this application during Oct 2011-Dec 2011. Sample Emages for the Flood Water Levels and the Shelter Coverage are shown in Figure 19. The central Bangkok city area was relatively well covered by shelters and also had low water level incisions because of walls built around the city. As can be seen from a sample result snapshot in Figure 20, large parts of country however were under severe threat (shown in red in the figure).

We wanted to use the information about the threat level to aid the people in severe risk level areas. As a first step we reached out to all people who had tweeted with a Flood related term in the last 24 hours. A sample Emage capturing the relative incidence of such tweets is shown in Figure 21[a]. Using the personalized action taking capability of EventShop we configured a rule which could automatically send back tweets to all the users in severe risk areas advising them to move to the nearest open shelter immediately. The tweets also contained a pointer to a web URL containing physical address and other detailed information (e.g. current vacancy, phone number, directions) about the nearest open shelter. The twitter account used for sending out the Tweets was @SocLifeNetworks. Some of the tweets sent out are shown in Figure 21[b]. As can be seen some of our tweets were re-tweeted by the receivers, thus indicating a positive interest in receiving and spreading such information.

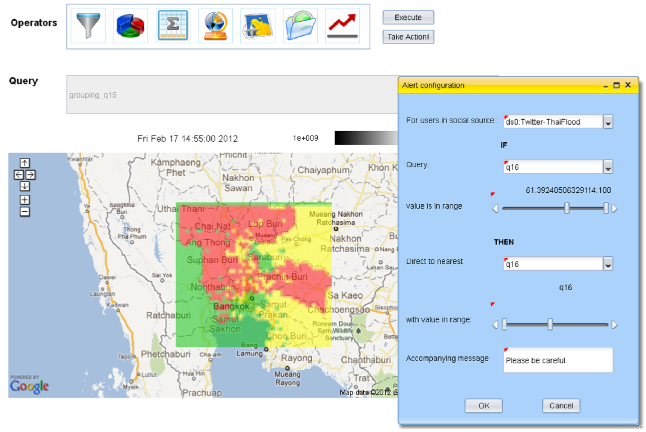

Figure 20: Classification on 'Flood threat level' and configuration to send out tweets.

Figure 21: (a): Sample Emage showing activity areas for tweets related to Thai Flood, and (b): Sample tweets sent out to real users in high risk situations.

6.4 Discussion and Future Outlook

We applied this framework to many applications, but presented here our experience with three. The three applications discussed were to demonstrate different aspects of the framework. We saw the complete process of modeling and detecting *Epidemic outbreaks*. In *Wildfire* detection application, we saw how such models can be revised and augmented to include diverse sources until satisfactory level of recognition performance is achieved. Lastly, in the *Thailand Flood Risk Recommendation* application we saw how such models can be used to build first-of-a-kind situation-aware applications which provide a complete loop from user generated data, to situation detection in the cloud, to alerts sent back to the users. By combining different data streams we were able to detect risky situations and aid the users affected in real time.

Put together, the three applications also highlight the adaptability and the expressiveness of the modeling approach and the framework at handling different situations across different spatio-temporal bounds in diverse application domains.

As mentioned, the video captures of testing the situation models using EventShop are available at [29]. Additionally [29] also provides video captures for a Hurricane mitigation application (which directs people to nearest shelters based on predicted hurricane path, population, and open Red Cross shelters data), and an Asthma/Allergy recommendation application (which detects Allergy Risk level based on combination of pollen count data, pollution level, and number of Twitter reports mentioning allergy symptoms).

Referring back to Figure 1, note that building the applications described required a situation driven perspective (and a relevant computational framework). The concepts detected were intrinsically evolving, occurred in the real world, and inherently manifested themselves over heterogeneous multimedia streams coming from multiple sources. An ability to combine data over both space (e.g. millions of search logs across locations) and time (e.g. historical average comparisons) was critical for handling these situations. Similarly sourcing the data from any relevant media type (incl. Twitter, Census, Satellite, Google Search logs, citizen-generated KML) was pivotal to modeling and solving the problems in the real-world (and not the media silos).

This paper shows the potential of Situation Recognition in aiding diverse human applications. Solving the variety of issues associated with Situation Recognition shall require a concerted community effort on multiple aspects like multimodal data fusion, scalable data analysis, data representation, media processing techniques, machine learning, and predictive modeling. This provides newer challenges (and opportunities) for the research community to work towards tackling each of these problems from a new situation-driven perspective. The rewards associated with Situation Recognition would clearly be unprecedented. We shall be able to maintain an evolving pulse of the world respond to various situations in real-time to save human lives and resources.

7. CONCLUSIONS

This paper motivates and computationally grounds the problem of combining heterogeneous dynamic big multimedia data into actionable situations. Specifically the paper focuses on describing a generic approach for modeling and recognizing situations. Looking back at section 1, this paper counters three fundamental problems in situation recognition. It provides a computational definition to the notion of situation. It presents a methodology for modeling situations based on generic conceptual blocks, and describes a toolkit to rapidly implement, validate, and refine these situation models. Results obtained across different applications highlight the potential of such an approach at detecting diverse situations. Further growth in the area of situation recognition is imperative, and would allow for detection of an evolving pulse of the world by combining heterogeneous, spatially-aware, real-time, big multimedia data.

8. REFERENCES

[1] Peter Diamandis, Coming Age of Abundance, TED Talk 2012.

[2] Withheld for blind review.

[3] Withheld for blind review.

[4] Endsley, M.R. (1988). Situation awareness global assessment technique (SAGAT). Proceedings of the National Aerospace and Electronics Conference (NAECON), 789–795. New York: IEEE.

[5] Moray, N. (2004). Ou sont les neiges d'antan? ("Where are the snows of yesteryear?"). In D.A. Vincenzi, M. Mouloua & P.A. Hancock (Eds), Human performance, situation awareness and automation: Current research and trends (pp. 1–31)

[6] Adam, E.C. (1993). Fighter cockpits of the future. Proceedings of 12th IEEE/AIAA Digital Avionics Systems Conference (DASC), 318–323.

[7] Jeannot, E., Kelly, C. & Thompson, D. (2003). The development of situation awareness measures in ATM systems. Brussels: Eurocontrol.

[8] J. McCarthy and P. Hayes (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, Machine Intelligence, 4:463–502. Edinburgh University Press, 1969.

[9] Stephen S. Yau and Junwei Liu, Hierarchical situation modeling and reasoning for pervasive computing, Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, 2006.

[10] The Situation Underground, Barwise and John Perry,Stanford Working Papers in Semantics, vol. 1, Stanford Cognitive Science Group 1980.

[11] Dietrich, D. Kastner, W. Maly, T. Roesener, C. Russ, G. Schweinzer, H., Situation modeling, IEEE Workshop on Factory Communication Systems, 2004.

[12] Sarter, N. B. and Woods, D. D. (1991). Situation Awareness: A critical but ill-defined phenomenon. International Journal of Aviation Psychology, 1, 45-57.

[13] Dominguez, C. (1994). Can SA be defined? In Situation Awareness: Papers and Annotated Bibliography (U). Armstrong Laboratory, Wright-Patterson AFB: OH.

[14] Smith, K., and Hancock, P. A. (1995). The risk space representation of commercial airspace. In Proceedings of the 8th International Symposium on Aviation Psychology, 1995.

[15] Dostal, B.C. (2007). Enhancing situational understanding through the employment of unmanned aerial vehicles. Interim Brigade Combat Team Newsletter, No. 01–18.

[16] V. K. Singh and R. Jain, Situation-based Control for Cyber Physical Environments, IEEE workshop on situation management, MILCOM, (IEEE SIMA), 2009

[17] Kaufman, Y. J., Justice, C. O., Flynn, L. P., Kendall, J. D., Prins, E. M., Giglio, L., et al. (1998). Potential global fire monitoring from EOSMODIS. Journal of Geophysical Research, [Atmospheres], 103, 32215– 32238.

[18] Jakobson, G. Buford, J. Lewis, L., A Framework of Cognitive Situation Modeling and Recognition, Military Communications Conference, 2006. MILCOM 2006

[19] Gísli R. Hjaltason and Hanan Samet. 1998. Incremental distance join algorithms for spatial databases. In Proc. ACM SIGMOD Int. Conf. Management of Data (SIGMOD '98).

[20] A trillion points of data, Barett Sheridan, Newsweek 2012.

[21] Steinberg, A.N., Bowman, C.L., and White, F.E., "Revisions to the JDL Data Fusion Model", in Sensor Fusion: Architectures, Algorithms, and Applications, Proceedings of the SPIE, Vol. 3719, 1999

[22] Oliver Brdiczka, Pong C. Yuen, Sofia Zaidenberg, Patrick Reignier, James L. Crowley, Automatic Acquisition of Context Models and its Application to Video Surveillance, International Conference on Pattern Recognition, 2006.

[23] R. Cucchiara, A.Pratti, and R.Vezani, "An Intelligent Surveillance System for Dangerous Situation Detection in home Environments" , in Intelligenza artificable, vol.1, n.1, 2004.

[24] J.D. Ullman, Principles of Database and Knowledge-base Systems, vol. 1, Computer Science Press, 1988.

[25] Montanari, M., Mehrotra, S., and Venkatasubramanian, N. 2007. Architecture for an automatic customized warning system. In IEEE Intelligence and Security Informatics, 2007

[26] Singh, K.D., 1993, The 1990 tropical forest resource assessment, Unasyva/FAO, 44, 10-19.

[27] A. K. Dey. Providing Architectural Support for Building Context-Aware Applications, PhD thesis, College of Computing, Georgia Institute of Technology, 2000.

[28] http://eventshop2012.appspot.com/?refresh=forced

[29] http://eventshop2012.appspot.com/videos/

[30] Asaf Adi , Opher Etzion, Amit - the situation manager, The VLDB Journal — The International Journal on Very Large Data Bases, v.13 n.2, p.177-203, May 2004

[31] P. D. Costa, J. Almeida, L. Pires, and M. Sinderen. Situation specification and realization in rule-based context-aware applications. International conference on Distributed applications and interoperable systems, 2007.