SATViewer - Visualizing A Large Number of Sensor Captured Streams on a Limited Display

Chiara Chiappini,
Department of Electronic Engineering
University of Bologna

Ronen Vaisenberg, Sharad Mehrotra and Nalini Venkatasubramanian
School of Information and Computer Sciences University of California, Irvine
Emails: ronen@uci.edu, sharad@ics.uci.edu, nalini@ics.uci.edu

I. Project Description

A. Introduction

Emerging multimodal sensing, embedded computing, and networking technologies have created opportunities to blend sensing and computation with physical spaces resulting in instrumented pervasive spaces (IPS). Sensor data captured in such environments can be used to model the state of the physical world, which, in turn, can be exploited for tasks such as information customization, building automation, inventory management and access control. We have created such a campus-wide instrumented pervasive space at UC Irvine called Responsphere which is fitted with variety of diverse sensors including video, audio, RFID, people-counters and environmental sensors to monitor temperature, humidity, concentration levels of atmospheric gases such as O2, CO etc. In this project, we focus on applications that require sensor data to be captured and stored over a period of time which can then be used for other purposes. For instance, we have developed a "people monitor" application that allows users of the IPS to reveal information about themselves, their location and availability in a limited way to others to help improve efficiency at work. A "space monitor" application can reveal how building spaces are utilized over time to restructure activities (e.g. meetings) to make better use of space.

Building such applications requires (1) methods for capture, storage and management of multimodal sensor data and (2) techniques to query, retrieve and visualize the stored data meaningfully. To address the former, we have developed SATware[3], a multimodal sensor data stream capture, analysis, and transformation middleware that aims at realizing a sentient system. SATware provides mechanisms the ability to capture multimodal sensor events and data, techniques to map application level events to basic media events detectable directly over sensor streams and a run-time for detection and transformation of multimodal sensor events and eventually storage of the annotated sensor data/events in a SATware database. This project focuses on the latter issue - i.e. retrieval and visualization of stored sensor streams.

B. SATViewer

Specifically, we describe SATViewer, which visualizes information captured in the SATware database which stores data collected from multiple sensors in the Responsphere IPS. The key challenge in designing a visualization tool for a pervasive system is that of *information overload* - limitations in user perception and in available display sizes prevent easy assimilation of information from massive databases of stored sensor data. For instance, in our setting, we have over 200 camera sensors deployed at two buildings; even a very simple query interested in monitoring these buildings will have to visualize 400 streams (audio/video) for any given time. In SATViewer,we have attempted to address the information overload problem using two key strategies – (a) ranking and prioritization of relevant sensor streams and (b) summarization of selected sensor streams.

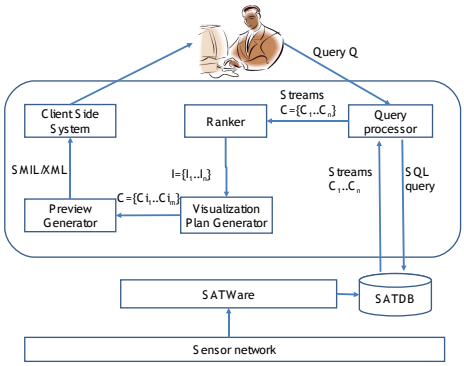


Fig. 1. SATViewer architecture

Given a user's query and the corresponding set of output streams that satisfy the query, the SATViewer handles the visualization of the query results based on the user's display limitations (e.g., number of display window, etc.). User's pose queries in a stream-based language SATQL that is similar to CQL[1] and the visualization plan is represented in XML using SMIL format[2]. Figure 2 illustrates a sample query to retrieve sensor streams from a building between 8 to 11am on a particular day. The user has further specified a function **activity()** to rank the streams specifying the basis for ranking the output streams. Relevant work was done by IBM in a system named S3 [4]. The proposed project is different as we visualize multiple events at the same time, and support high level events and queries.

Select all streams
Where Bren Hall building
When 4-4-2008 8:00 a.m. and 4-4-2008 11:00 a.m.
Ranked by activity()

Fig. 2. A Sample Query

II. Students Research-Related Duties and Expected Outcomes

SATDB: is the sensor database that stores the sensor streams. Student will add support for high level events to the SATDB; In addition to storing raw sensor data, the outcome of the student's work will allow it to also store higher level event information derived from the sensor data. For instance, it may annotate the sensor data with activities and / or events as well as identity of the person involved if the corresponding operators for such analysis were incorporated over the stream data. Irrespective of the nature of data (i.e., raw sensor data or events)

SAT Ranker: In general, the query predicates might not be precise. For instance, if automated models are used to derive semantic annotations with the streams (e.g., presence of a certain person) and the query contains a selection predicate on such annotations, the resulting streams we may not be able to determine precisely that the resulting substream "matches" the specified query. In such a case, the system can determine the degree to which a stream satisfies the input query and use that for the purpose of ranking the streams. The output of SAT ranker is the set of answer streams annotated with a corresponding ranking based on the ranking function.

Student will add support to an existing ranker, such that it can select high level events.

Visualization Plan Generator: takes the set of ordered answer streams as well as display parameters of user's display to generate a SMIL plan which determines a Visualization Plan. Visualization Plan Generator selects streams to display and which streams to leave out, based on the display limitations.

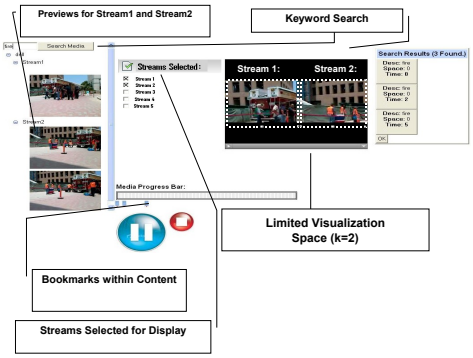
Choosing the streams to display is the well known weighted interval selection problem, when multiple resources are available. In our case, K resources, the number of players available.

Student will add support for visualization of high level events.

Preview Generator: The user might want to browse the content for perhaps skip parts until "interesting" content is displayed. For this reason, the streams undergo a final processing step, where the system will **generate relevant previews** for the different streams based on the changes in the streams. Student will add support for high level events.

A. Student Eligibility - Technical Information about the Project

The implementation of the project is in JAVA, utilizing Google's GWT framework on top of a database. Student has to be fluent in JAVA, JAVASCRIPT and SQL. Following are some more technical details of the system: **Client Side Component:** which enables the visualization of query results, previews, bookmarks and search using an XML standard which is an extension of SMIL[1]. the Visualization plan is visualized by a regular web browser using a SMIL plug-in.



(a) SATViewer

```
<xml>
<smil>
<head><layout>
 <region id="region1"/>
 <region id="region2""/>
</layout></head><body>
 <par><seq>
<video src="S1.mp4"
 clip-begin="08:00:00:00"
 clip-end="9:00:00:00"
 region="region1 "/>
<video src="S3.mp4"
 clip-begin="09:00:00:00"
 clip-end="10:00:00:00"
 region="region1 "/>
</seq><seq>
<video src="S2.mp4"
 clip-begin="08:00:00:00"
 clip-end="09:00:00:00">
region= "region2" />
<video src="S4.mp4"
 clip-begin="09:00:00:00"
 clip-end="10:00:00:00">
region= "region2" />
</seq></par></body>
</smil>
```

```
...
...
 <toc title="drill">
  <smilObject>
   <preview title="Stream1">
    <preview time="0"
image="file304.png">
    </preview>
    <preview title="Stream2">
    .....
    <bookmark title="John entered
the scene"
time="0">
</bookmark>
<tagList>
<term title="fire" time="0,2,5">
</term></tagList>
    ....
  </smilObject>
</toc></xml>
```

(b) XML Source

Fig. 3. A Demonstration of a Visualization of k=2 Streams and its Relevant XML Source

References

[1] A. Arasu, S. Babu, and J. Widom. The CQL continuous query language: semantic foundations and query execution. *The VLDB Journal The International Journal on Very Large Data Bases*, 15(2):121–142, 2006.

[2] J. Ayars, D. Bulterman, A. Cohen, K. Day, E. Hodge, P. Hoschka, E. Hyche, M. Jourdan, K. Kubota, R. Lanphier, et al. Synchronized Multimedia Integration Language (SMIL) 2.0 Specification. *Work in progress. W3C Working Drafts are available at http://www. w3. org/TR*, 21, 2000.

[3] B. Hore, H. Jafarpour, R. Jain, S. Ji, D. Massaguer, S. Mehrotra, N. Venkatasubramanian, and U. Westermann. Design and Implementation of a Middleware for Sentient Spaces. *Intelligence and Security Informatics, 2007 IEEE*, pages 137–144, 2007.

[4] C.F. Shu, A. Hampapur, M. Lu, L. Brown, J. Connell, A. Senior, and Y. Tian. IBM smart surveillance system (S3): a open and extensible framework for event based surveillance.

[1]SMIL[2] is a W3C standard which allows users to create multimedia presentations on the world wide web. SMIL provides presentation authors with the ability to control the timing of the presentation and placement of presentation objects in the presentation document.