Securing Critical Infrastructures via Geometric Modeling and Discrete Simulation

[3]*P. Assogna,* [2]*G. Bertocchi,* [1,4]*A. Paoluzzi,* [1,4]*G. Scorzelli,* [4]*M. Vicentino, and* [3,4]*R. Zollo* [*]

[1]Dept. Informatics and Automation, “Roma Tre” University, Rome, Italy
[2]Master on Security and Protection, University “La Sapienza”, Rome, Italy
[3]Theorematica spa, Rome, Italy
[4]TRS srl, Rome, Italy

Abstract

Next generation of awareness and security systems must integrate pre-existing subsystems with novel capabilities, including: virtual-reality and advanced interfaces, behavior and situation modeling, and data mining, on the common basis of 3D modeling and simulation. Through modeling and simulation of known behaviors, such a system may recognise the weak signals of risky situations, and promptly activate the countermeasures. Awareness requires knowledge: the infrastructure, the users, the resources, the events, must be dynamically modeled and simulated in their relevant aspects. The information on the infrastructure and its environment will be analyzed to determine the events dynamics as reported by networks of sensors (video-surveillance, movement, fire alarm, etc). Further knowledge is added by feed-back from the operators. Using both simulated and user-fed knowledge, an advanced system should also provide, in our vision, both training and decision-support tools. Such an advanced system is being developed to supply the security operators with optimized planning of normal operations and what-if simulation for emergency planning and crisis management.

---

[*] p.assogna[at]theorematica[dot]it, bertocchi_g[at]camera[dot]it, paoluzzi[at]dia[dot]uniroma3[dot]it, scorzell[at]dia[dot]uniroma3[dot]it, michele[dot]vicentino[at]libero[dot]it, roberto[dot]zollo[at]theorematica[dot]it

1 Introduction

It is well known and widely recognized that *One Look is Worth A Thousand Words*, so it may really provide a higher-level of situation awareness for security and safety. Accordingly, we show here that an advanced geometric platform, providing a shared digital model of the system to be protected, and combined with a concurrent language for event discovery and tracking and situation evaluation, is the best candidate to: serve as point of reference for the integration of vision, sensor, tracking and security systems committed to infrastructure protection; provide a reliable basis for high-level situation awareness; enable coordinated and optimized decision-making.

In particular, critical transport infrastructures, especially when crossing national borders (such as tunnels, bridges, railways hubs, airports, etc), require a novel security approach. Their security, i.e. the capacity of preventing threats and reacting to menaces, should be based on a strong coordination of daily operations, since a security threat can arise not only from a malicious attack but also from natural events (storms, floods, etc.) or unexpected facts, like traffic congestion or a collision. Security and safety should be maintained under permanent control, and the security systems should be able to deal with both problem facets. Moreover, at transnational level, like within Europe, there is also a compelling need for a coherent and coordinated behavior of national agencies, not only for emergency management but also for daily operations.

For this purpose, an advanced security system needs to infer the consequences of events from information received, where some missing information may be even regarded as present. This is the actual value of virtual/augmented reality and advanced interfaces [Sebe *et al.*, 2003, Ott *et al.*, 2006], in our view. Let consider, as an example, a gallery full of smoke, or a railroad under a great storm, where video does not work, or a fault of energy to a sensor system. An advanced security system should use virtual/augmented reality to reproduce the situation to be controlled, and this ability consequently allows for optimized decision-making through modeling and simulation. These requirements can be summarized as a strong necessity for permanent awareness, that means having continuous knowledge, perception and understanding of the situation. Conversely, present-day security systems can be generally viewed as clever assemblies of sensor subsystems, with very limited capabilities of assisting the personnel during normal operations and crises; moreover, they are not designed to be used when the responsibility is shared between different control and security organizations, as it happens referring to critical infrastructures at transnational level.

In this paper we discuss the development goals and the implementation directions of a new platform for security of critical infrastructures being developed with a research grant from the Italian Ministry of University and Research (MUR) to TRS, which stands for "Technology and Research for Security", a spinoff company from University Roma Tre and Theorematica. The main ideas of this project can be summarized as follows: (a) (mostly) automatic 3D modeling of the infrastructure; (b) 3D integration of sensor networks, including videosurveillance; (c) situation modeling though hierarchical event graphs annotated with actor models and behaviors; (d) visual simulation via 3D animation; (e) situation analysis based on the divergence of sensor feedback from evolution of simulations; (f) weak signals of risky situations deduced from comparison of situation analysis with a knowledge base of past events and situations. Such very sophisticated modeling, simulation and analysis are permitted by the use of powerful computational tools for geometric modeling and concurrent programming, discussed in the remainder of this paper.

We plan the first version of the platform ready in one year, and share the main concepts in the current contribution, with identification and discussion of what we consider the proper tools. In particular, the use of an Erlang-based implementation of a functional geometric language. We discuss the huge advantages in using such an architecture for implementing a security platform. Erlang may start two hundred thousand processes in a fifth of a second; Plasm may generate on-the-fly the model of the leaning tower of Pisa in 0.7 secs on a cluster of 16 pentium nodes, starting from a 14K source file, and 80 node silicium chips are at the door... In order to perform the discrete simulations needed by the next generation of security systems we strongly need such a descriptive power and the complexity control provided by the discussed platform. We believe that the extreme complexity of the simulations needed to enforce the security of critical infrastructures cannot be approached by using the Von Neumann architecture [Backus, 1978] and the languages at value-level that stand on it.

2 A new paradigm for awareness

Present security systems do not include the high-level functionalities needed to accomplish complex security tasks, that do not consist only in giving alerts and alarms based on signals from a sensor network, but mainly in (a) noticing any unusual and potentially dangerous or difficult circumstances, (b) recognizing the weak signals of risky situations, (b) promptly activating the needed countermeasures, and (c) assisting the personnel in optimized management of the crisis.

This gap between available solutions and actual requirements for security of critical infrastructures requires a new generation of systems able to integrate existing subsystems with beyond the state-of-the-art capabilities like: virtual reality user-interfaces, 3D modeling as reference paradigm, behavior and situation modeling, mining of knowledge and data bases, etc. A new type of awareness platform can be therefore conceived by loose but effective integration of:

1. 3D geometric models of the infrastructure, to be used as reference base for every collected information, and built by novel and greatly automated techniques;
2. virtual and augmented reality interfaces and serious gaming advanced techniques;
3. an interoperable knowledge framework, derived by models of: the infrastructures, the normal behaviors, the events, the users, the resources, to be used as base for statistical and situational models;
4. the pre-existing heterogeneous sensor and security systems.

Through the integration of the previous components, a next generation security platform should deploy: sensor fusion, abnormal behavior detection, event discovery and tracking, support for the evaluation and optimization of decision making, “what-if” simulation capabilities.

The surveillance of a critical infrastructure cannot limit itself to a representation of what is happening, but has to provide tools for analyzing the patterns and rhythms and interplays of the events, and for anticipating, as much as possible, future impacts. In this way the response of the control resources (both automatic and human) can be more effective and proactive. An important abstraction of the security platform will concern the *Software Agents* performing all the activities of sensing the environment, acting on devices, alerting security personnel, and so on. In particular we imagine an holonic, multi-level organisation of these custodians, which will maintain and use an awareness base represented by the integration of models, of environment sensing, of surveillance and control activities. A *holon* [Koestler, 1967] is the component of a system that can work autonomously even if detached from it, and that works well in cooperation with other holons. In this vision the security platform should represent the intelligence of the controlled infrastructure, like a living organism endowed with self-consciousness [Koestler, 1967].

For this purpose the introduced platform maximally enforces the modeling capabilities, because all supports provided are grounded on model-based simulations. Permanent situation awareness will be obtained through techniques for modeling the infrastructure, simulating its normal behavior, acquiring and tracking both normal and abnormal events, understanding the weak signals introducing a crisis and reacting to them in the best way. Through the capability of simulating events, actions and reactions, the security platform will maintain the critical infrastructure in a controlled and safe state. The operating personnel will interact adding the experience and judgment that cannot be coded or simulated.

Notice that security people will be in the loop, differentiating this approach from a claim of autonomic computing. With respect to how the problem of false positives will be addressed, a major issue when seeking to model normal and abnormal events, our current idea is to use the divergence of reality from simulations to set a higher alert level, to increase both the sensor resolution and the grain of simulations, and to loop until the human operator is alerted. Each security incident will contribute to add new *event genotypes* ("internally coded, inheritable information") to the knowledge base, to be compared with future *event phenotypes* ("outward, physical manifestation"). But such an approach must be implemented and checked on the field. Several machine learning techniques will be tested at this purpose.

To reach such objectives, a novel technology for symbolic situation modeling, based on machine- and human-readable, executable, and combinable storyboards is being developed [Assogna *et al.*, 2008] using a very high-level concurrent programming language. For this purpose we are making experiments with the Erlang programming language [Armstrong, 2003, Armstrong, 2007], formerly developed at Ericson, integrated by an advanced environment for symbolic 3D modeling based on the geometric language Plasm [Paoluzzi, 2003, Corney, 2004, PLaSM Web Site, *a*], and with some engines for serious gaming. Such concurrent environment will encode, simulate, animate, and interact with:

1. security protocols and procedures;
2. geometric models of the infrastructure and its surrounding environment;
3. sensor data;
4. recognized events;
5. normal and abnormal behaviors;
6. communities of software agents and human operators.

Our modeling and simulation environment will also allow for visual combination of existing programs into more complex ones, and will support situation simulation for decision making, using serious gaming and advanced interface technology.

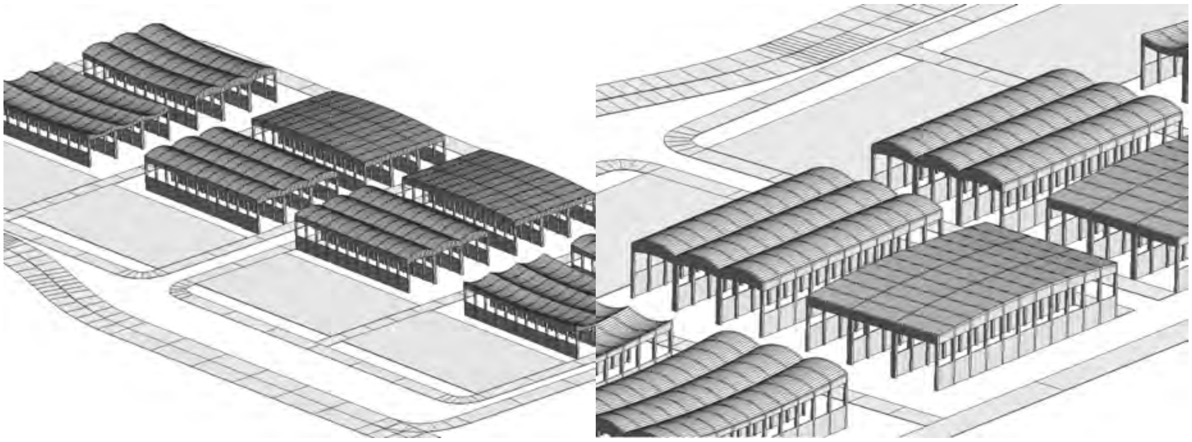

Figure 1: Virtual model of a commercial district.

3 Infrastructure and event modeling

In his essay entitled “Spatial Models for Wide-Area Visual Surveillance: Computational Approaches and Spatial Building-Block” [Howarth, 2005], the author demonstrates that spatial models play a key role when interpreting a dynamic and uncertain world for a surveillance application. In particular, the paper illustrates the range of spatial models by surveying various works relevant to the development of spatial models, and how they have been used in AI applications of security. Then a cellular decompositive representation of the space is chosen as the most useful foundation to support visual surveillance applications.

This one is the basic element, associated to the BSP (Binary Space Partition) spatial index for fast point location [Fuchs *et al.*, 1980, Naylor *et al.*, 1990], of the advanced representation scheme already used by the geometric language PlaSM (the Programming LAnguage for Solid modeling) [Paoluzzi, 2003, Paoluzzi *et al.*, 1995]. The paper [Howarth, 2005] also discusses the need of associating a semantics to the hierarchical elements of the spatial subdivision. Fortunately, such a purpose is already perfectly performed by the symbolic descriptive approach provided by a geometric language.

For the symbolic support to the generation and advanced handling of the geometric information, we therefore use the geometric language PlaSM [PLaSM Web Site, *a*, Corney, 2004]. The run-time evaluation of the sub- models of building portions interested to each protection procedure or training simulation will provide the 3D scenes to be used at the operation and control centre. Several multiplatform tools are available for PLaSM programming, including a visual editor (Xplode) and an IDE (Integrated Development Environment) plug-in for the industrial-strength open development platform Eclipse [Holzner, 2004]. PLaSM Version 5 being currently deployed, is in our opinion the best and more advanced platform for spatial modeling and reasoning. In Figure 1 we show the virtual model of a warehouse district produced by a couple of pages of Plasm source code.

Language basics The design language PLaSM is a geometry-oriented extension of a subset of FL (programming at Function Level), a functional language developed by Backus’ group at IBM Research Division in Almaden (USA) [Backus *et al.*, 1990, Aiken *et al.*, 1991]. On the line of [Backus, 1978] Turing lecture, FL introduces an algebra over programs, and enjoies several interesting features. In particular, programs are easily combined, so that new programs are obtained in a simple and elegant way; and one may find simpler equivalent programs, both at design and at compilation times. Great advantages are so obtained in style and efficiency of program prototyping.

The PLaSM language is extensible by design. Even the most common graphics primitives can be natively defined in PLaSM, and novel geometric operations can be easily added to the language by adding them in a library. The current libraries contain about 600 functions. For an introduction to the language, the reader is referred to [Paoluzzi, 2003].

Primitive objects are characters, numbers, truth values and *polyhedral complexes*. A polyhedron is a quasi-disjoint union of polytopes, i.e. of bounded convex sets. *Expressions* are either primitive objects, functions, applications or sequences. According to the FL semantics, an arbitrary PLaSM script can be written by using only three programming constructs:

application `f:x` of a function `f` to the actual value `x` of the input *parameter*, element of the function domain, producing an output *value* in the function codomain;

composition of two or more functions: `(f∼g∼h):x` ≡ `(f∼g):(h:x)` ≡ `f:(g:(h:x))`, producing the pipelined execution of their reversed sequence (see Figure 2);

construction of a vector function `[f,g,h]`, allowing for the parallel execution of its component functions `[f, g, h]:x` ≡ `<f:x, g:x, h:x>`.

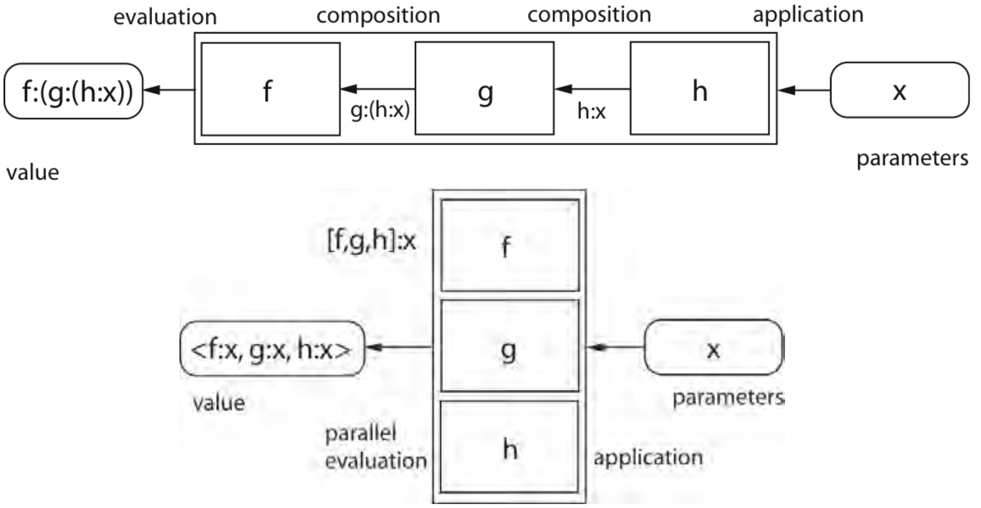

Figure 2: Visual representation of the main paradigms of the Plasm language:
(a) pipelined execution — `(f ∼ g ∼ h):x` ≡ `(f∼g):(h:x)` ≡ `f:(g:(h:x))`;
(b) parallel execution — `[f, g, h]:x` ≡ `<f:x, g:x, h:x>`.

3.1 Model generation

For the development of the project discussed here, we capitalize on a novel parallel framework [Bajaj *et al.*, 2006, Scorzelli *et al.*, 2007] for high-performance solid and geometric modeling, that (i) compiles the generating expression of the model into a dataflow network of concurrent threads, and (ii) splits the model into fragments to be distributed to computational nodes and generated independently. Progressive BSP trees are used for adaptive and parallelizable streaming dataflow evaluation of geometric expressions [Scorzelli *et al.*, 2007], and associated to the polyhedral cells of the HPC (Hierarchical Polyhedral Complex) data structure [Pascucci *et al.*, 1995] used by the language. Hasse graphs are used to maintain a complete representation of the model topology. A novel tensorial representation of the chain complex mock-up of the model, that we called Hasse matrix [DiCarlo *et al.*, 2007], may allow to support both a geometric design and multiphysics simulations over it.

From plans to solid models The security platform will make strong use of spatial models produced as cellular decomposition of structures (buildings, tracks, lanes, tunnels, bridges, etc.) from 2D plans. We are consequently working on automatic generation of PLaSM models from line drawings in Autocad format, that is the only kind of geometric information widely available [Assogna *et al.*, 2008]. A geometric representation of whatever infrastructure portion may be of interest, progressively generated at increasing level-of-detail, will be produced at runtime by a streaming dataflow process. We may consider implementing it on the Cell/B.E. architecture [Williams *et al.*, 2006, Gschwind *et al.*, 2006], whose costs are rapidly moving down. Furthermore, the native use of a BSP tree [Fuchs *et al.*, 1980, Naylor *et al.*, 1990] allows for fast point location, i.e. the fast translation of a triple of coordinate numbers, possibly GPS-generated, into its location within the proper infrastructure part and position, as well as for dynamic level-of-detail in realistic visualization of complex environments.

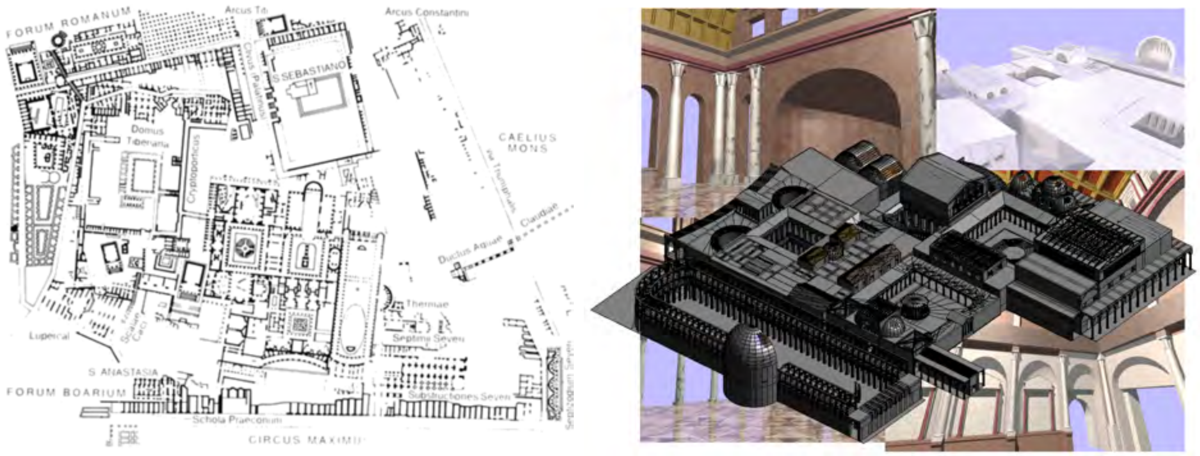

Figure 3: (a) 2D map of the Palatino hill in Rome; (b) the 3D Plasm reconstruction of the emperor's palace (Domus Augustana), a site endowed with the strongest security system of the ancient times. Notice the different awareness levels conveyed by 2D and 3D models.

The generation of models of complex buildings from plans is a difficult and largely unsolved inverse problem [Lewis & Séquin, 1998]. The difficulty of the reconstruction problem of 3D buildings from architectural plans is confirmed by the rarity of scientific literature on the topic (see [Lewis & Séquin, 1998, So *et al.*, 1998, Teller, 2003, Howarth, 2005]). A fast semi-automatic solution [Assogna *et al.*, 2007] was already introduced, that can be summarized as follows [Paoluzzi & Scorzelli, 2007]. First of all, input Autocad files with line-drawings of 2D architectural plans are transformed into double interval trees and quad-trees, to efficiently answer interactive proximity queries. Then semantics is assigned to small line subsets, via pattern-based recognition of the components of building fabric (internal partitions, external enclosures, vertical communication elements etc), and with subsequent translation into PLaSM scripts, i.e. symbolic generating forms. Later, the evaluation of symbolic scripts may produce either streaming solid models at variable levels of detail or adjacency graphs of the critical infrastructure as a whole or of parts of the building that are interested to some specific security or protection procedure.

4 Situation simulation

Modeling techniques tend to be found in isolated communities [Park & Fishwick, 2005]: geometry models in computer-aided design (CAD) and computer graphics, dynamic models in computer simulation, and information models in information technology. When models are included within the same digital environment, the ways of connecting them together seamlessly and visually are not well known. A main challenge of our project will concern the discover of the best abstractions to integrate such three different modeling areas.

According to [Miller *et al.*, 2004], in modeling and simulation of discrete concurrent systems three paradigms are used, namely event scheduling (event graphs and PERT), activity scanning (bipartite graphs and Petri Nets), and process interaction (finite-state systems and Markov chains).

To animate a complex simulation, network programming techniques [Bajaj *et al.*, 1999] may be used, where the behavior of each actor is described by a path of an activity network, that codifies the storyboard of affairs as a DAG (Direct Acyclic Graph), and describes the causal and temporal relationships between events. Furthermore, each storyboard arc may be associated to a spline in the configuration space of the actor, that models the fine behavior of its degrees of freedom. This model is hierarchical, since each arc may be substituted by its (local) storyboard, used to decompose a macro event to a finer level-of-detail, as well as to compute the parameters of the probability distributions of the stochastic variables of interest (e.g. the most likely time or the lead/lag time of an event). The depth of the event hierarchy may vary locally, depending on the complexity of the events and on degree of realism of the simulation [Paoluzzi & D'Ambrogio, 1999].

Petri nets [Peterson, 1977, Peterson, 1981] may be used to mathematically describe the evolution of a concurrent system, using networks with two types of nodes (places and transitions) and arcs where several types of tokens may move on the network. The state of the system is given by the distribution of tokens in the places. The annotations of fine behavior are again described by splines in configuration space, defined by discrete sets of points consisting at least by two known configurations, when a linear behavior is given. Petri nets may be hierarchical and timed. The HCSM framework (Hierarchical, Communicating and concurrent State Machine) is used to plan the behavior of reactive synthetic agents in interactive simulation environments [Cremer *et al.*, 1995]. The HCSM framework is adequate to model the reaction of autonomous agents and to direct those to produce desired situations.

Surveillance systems consist of three main elements: Data acquisition, Information analysis, and On-Field operation. Recent, large surveillance systems acquire data from hundreds of networked cameras [Ott *et al.*, 2006]. With the increase of the number of cameras and other sensor-data, the information analysis becomes increasingly difficult. Human operators can easily be overwhelmed by this flood of unorganized visual information, and fail to effectively understand the situation and properly inform On-Field operations [Sebe *et al.*, 2003]. First, studies indicate that human factors in the information analysis process strongly influence on the overall effective and efficient operation of the surveillance system [Smith, 2004]. Second, although (semi-) automated tracking and detection procedures are being put to use in daily practice, the need remains for an integrated situational understanding.

Also, the use of simulators and serious games [Löffler *et al.*, 2007, Jenvald *et al.*, 2007] for training and evaluation scenarios are applied in security situations [Kruchten *et al.*, 2007, Losh, 2007], and will be strongly enforced with high-level tools for situation evaluation in the project described here.

5 System architecture

We are developing a prototype platform for centralized situation awareness and analysis. The main concept at the base of this project is the integration of advanced modeling, simulation, virtual and augmented reality, serious gaming and tracking techniques in one centralized framework, where the tasks involved in maintaining a critical infrastructure in a safe state can be evaluated and optimized, and where the security personnel is provided with high-level decision-making tools, including interoperable situation modeling and simulation.

5.1 Concurrent programming

The security system discussed in this paper is being implemented using a combination of two functional languages: *Plasm* for geometric programming and *Erlang* for simulation based on concurrent programming. The geometric modeling and simulation aspects of the project were discussed in a previous section. The Erlang programming is introduced here.

Erlang [Armstrong, 2003, Armstrong, 2007] is a concurrent functional programming language and runtime system. It is characterized by strict evaluation, single assignment, and dynamic typing. Strict, or eager evaluation, means that expressions are evaluated as soon as they are bound to a variable, and the value is subsequently recalled without evaluation. The language is purely functional, without mutable state (induced by multiple assignment), and so the code is easy to understand and to prove for correctness. Dynamic typing means that type checking is performed at run-time.

For concurrency, Erlang follows the mathematical model of computation that uses "actors" as the universal primitives of concurrent computing [Hewitt *et al.*, 1973]. The *actor model* has been used both as a framework for a theoretical understanding of concurrency, and as the theoretical basis for several practical implementations of concurrent systems. An actor can make local decisions, create other actors, send and receive messages, and determine how to respond to received messages. Data structures, functions, semaphores, monitors, ports, descriptions, logical formulae, numbers, identifiers, demons, processes, contexts, and data bases can all be shown to be special cases of actors, whose behaviors can all be described by message sending.

Erlang was designed by Ericsson to support distributed, fault-tolerant, soft-real-time, non-stop applications. It even supports hot swapping of programs, so that code can be changed without stopping a system. The Erlang language is going to fit very well with the next generation of multi-core CPUs and Symmetric Multi-Processing (SMP) architectures, which operates on the principle that large problems can almost always be divided into smaller ones, that may be carried out concurrently, with several computational processes executing at the same time, and potentially interacting with each other. SMP systems allow any processor to work on any task no matter where the data for that task are located in memory; with proper operating system support, SMP systems can easily move tasks between processors to balance the workload efficiently.

In Erlang, processes belong to the programming language and not the operating system. The language possesses a number of extremely useful features [Armstrong, 2007] to be exploited within a security system: (a) creating and destroying processes is very fast; (b) sending messages between processes is very fast; (c) processes behave the same way on all operating systems; (d) we can have very large numbers of processes (up to tens of thousands); (e) processes share no memory and are completely independent; (f) the only way for processes to interact is through message passing.

5.2 Event Processing

Complex Event Processing (CEP) techniques, implemented using Erlang, will be used system-wide. The CEP integration architecture ([Wu *et al.*, 2006, Luckham & Frasca, 1998] and [Perrochon *et al.*, 1999]) is the state-of-the-art for stream event processing in complex computer systems. Different events, combined with different behaviors and environments, will elicit adequate responses from the Platform so that, even when unattended, it can act in order to avoid or minimize the unwanted scenarios. Optimal control will be achieved, of course, combining the automatic responses with the experience and judgment of security personnel.

This approach to security awareness, providing sensing fusion, as well as event discovering, registration and interpretation, will make strong use of spatial models produced as cellular decomposition of structures from architectural plans produced by semi-automatic generation [Assogna, 2005, Paoluzzi & Scorzelli, 2007] of PLaSM models from 2D line drawings, the only kind of geometric information widely available.

In our security framework, we will specifically focus on advanced techniques for interactive visualization of, and user interaction with, both real-time and recorded surveillance data. Knowledge from the fields of visual perception, 3D interactive computer graphics, Virtual Reality and Serious Games will be incorporated. To obtain a task-specific interface optimized for situational awareness we will integrate, apply and extend the state-of-the-art on 3D navigation and spatial montage of streaming sensor data (see [Snavely *et al.*, 2006]) .

5.3 Management supports

The platform will also provide several supports and operating aids to management of workforce, in case of crises that exceed its capability of automatic management. Here, tasks range from the daily routine of camera surveillance by operators to ad-hoc crisis management by controllers and decision makers. These various users need to access and interact with all information made available through the underlying databases and (generated) knowledge. Flexible, task-specific user interfaces are needed to allow for accumulation, retrieval and presentation of information.

We aim to design and evaluate a set of interface guidelines and components providing highly effective, task-specific user interface, enabling both automatic (SW agent-operated) and operator-controlled actions for security and safety monitoring, managed from an Operation Control Centre. In particular, the system will provide a semantically rich support to operating personnel based on the metaphors of newspaper, agenda, map, telephone and TV.

6 Conclusion and discussion

In this paper we have introduced some new concepts for situation awareness and higher security of critical infrastructures, including: geometric modeling and virtual and augmented reality, in order to provide the interoperation layer among security subsystems and between the platform and the security personnel; advanced user interfaces and serious gaming for semantically rich awareness of situation; behavior and situation modeling as support for simulations needed for decision-making.

From a more technological viewpoint, the novel security platform will rely on (a) novel multicore and multiprocessor architectures, on (b) very-high level programming supports for geometric modeling of the infrastructure and virtual reality interfaces, and (c) on a concurrent functional language for situation modeling, data mining and discovery of weak signals of risky situations. The approach is scalable with the complexity of the infrastructure. The geometry engine may produce on-the-fly the 3D model of the interested scene with progressive level-of-detail, and the distributed Erlang only need to be recompiled in order to use novel multicore and multiprocessor environments, including clusters.

Our paradigmatic reference for awareness, simulation and control of both normal and abnormal situations and behaviors, to be used for operation planning, security enforcing training and crises management, is to successful PLM (Product Lifecycle Management) platforms in aerospace, automotive and manufacturing industries, where the geometric information provides the exchange and collaboration layers among all business departments and all product data.

In order to provide the geometric integration engine of the new security platform, we are already implementing a novel Erlang-aware version of Plasm. The automatic reconstruction of 3D models from plans and the 3D integration of data streams from sensor networks, including video-surveillance, will provide the basic components of the platform. The subsequent step will concentrate on the development of a special-purpose simulation engine for events of variable granularity. Conversely, we intend to use open-source game engines and social software for the user interface and the training-oriented components. The development goals include high scalability and portability of the platform on next generation hardware architectures.

It is hard to discuss about the expected limitations of the approach in this stage. The plans are to define a modular, scalable, and extensible platform, offering a living environment for a system of security systems, including current and future sensing systems, peer-to-peer and hierarchical self-organization, self-perception, event capture and tracking as well as event modeling and simulation, geometric reasoning, and real-time visualization and interaction. Each point is challenging. Some technologies are well-established, some others are not. We must try and see ... Actually we are confident about our ability to semi-automatically produce high-quality models of very complex infrastructures, and to generate realistic interactive environments and real-time simulations, including feed-back from sensors in the protected area. The machine-learning and reasoning goals are obviously more uncertain and open to doubt.

Acknowledgements

We would like to thank the anonymous reviewers, whose comments greatly helped to make clearer the goals of the discussed approach. The research is funded by a seeding grant (art. 11 D.M n.593/2000) from MiUR (Italian Ministry of University and Research) with title "Sistemi Intelligenti per la Sicurezza delle Infrastrutture Critiche"

References

[Aiken *et al.*, 1991] Aiken, A., Williams, J. H. & Wimmers, E. L. (1991). The FL project: the design of a functional language. Unpublished report.

[Armstrong, 2003] Armstrong, J. (2003). *Making reliable distributed systems in the presence of software errors*. PhD thesis, The Royal Institute of Technology, Stockholm, Sweden.

[Armstrong, 2007] Armstrong, J. (2007). *Programming Erlang, Software for a Concurrent World*. Pragmatic Programmers.

[Assogna, 2005] Assogna, P. (2005). *Frattali di conoscenza*. Ed. ITER, Milan, Italy. In Italian.

[Assogna *et al.*, 2007] Assogna, P., Bertocchi, G., Gehrke, W., Milicchio, F., Paoluzzi, A., Portuesi, S., Scorzelli, G., Vicentino, M. & Zollo, R. (2007). VR platform for protection modelling of critical infrastructures. In *Virtual Reality International Conference (VRIC'07)*.

[Assogna *et al.*, 2008] Assogna, P., Bertocchi, G., Paoluzzi, A., Scorzelli, G. & Zollo, R. (2008). From 2D plans to 3D building models for security modeling of critical infrastructures. *International Journal of Shape Modeling,* **14** (1).

[Backus, 1978] Backus, J. (1978). Can programming be liberated from the von Neumann style?: a functional style and its algebra of programs. *Commun. ACM,* **21** (8), 613–641. (Turing Award lecture).

[Backus *et al.*, 1990] Backus, J., Williams, J. H. & Wimmers, E. L. (1990). An introduction to the programming language FL. In *Research topics in functional programming* pp. 219–247. Addison-Wesley Longman Publ. Boston, MA, USA.

[Bajaj *et al.*, 2006] Bajaj, C., Paoluzzi, A. & Scorzelli, G. (2006). Progressive conversion from B-rep to BSP for streaming geometric modeling. *Computer-Aided Design and Applications,* **3** (5 (6)). Preliminary version in CAD06, June 19-23, Phuket Island, Thailand (Best paper award).

[Bajaj *et al.*, 1999] Bajaj, C. L., Baldazzi, C., Cutchin, S., Paoluzzi, A., Pascucci, V. & Vicentino, M. (1999). A programming approach for complex animations. Part I. Methodology. *Computer-Aided Design,* **31** (11), 695–710.

[Corney, 2004] Corney, J. R. (2004). A PLaSM Primer: Geometric Programming for Computer-Aided Design. *Computer-Aided Design,* **36** (13), 1421–1422.

[Cremer *et al.*, 1995] Cremer, J., Kearney, J. & Papelis, Y. (1995). Hcsm: a framework for behavior and scenario control in virtual environments. *ACM Trans. Model. Comput. Simul.* **5** (3), 242–267.

[DiCarlo *et al.*, 2007] DiCarlo, A., Milicchio, F., Paoluzzi, A. & Shapiro, V. (2007). Solid and physical modeling with (co)chain complexes. In *SPM '07: Proceedings on the second ACM symposium on Solid and Physical modeling* ACM Press, New York, NY, USA.

[Fuchs *et al.*, 1980] Fuchs, H., Kedem, Z. M. & Naylor, B. F. (1980). On visible surface generation by a priori tree structures. *SIGGRAPH Comput. Graph.* **14** (3), 124–133.

[Gschwind *et al.*, 2006] Gschwind, M., Hofstee, H. P., Flachs, B., Hopkins, M., Watanabe, Y. & Yamazaki, T. (2006). Synergistic processing in cell's multicore architecture. *IEEE Micro,* **26** (2), 10–24.

[Hewitt *et al.*, 1973] Hewitt, C., Bishop, P., Greif, I., Smith, B., Matson, T. & Steiger, R. (1973). Actor induction and meta-evaluation. In *POPL '73: Proceedings of the 1st annual ACM SIGACT-SIGPLAN symposium on Principles of programming languages* pp. 153–168, Acm, New York, NY, USA.

[Holzner, 2004] Holzner, S. (2004). *Eclipse*. O'Reilly Media.

[Howarth, 2005] Howarth, R. J. (2005). Spatial models for wide-area visual surveillance: computational approaches and spatial building-blocks. *Artif. Intell. Rev.* **23** (2), 97–155.

[Jenvald *et al.*, 2007] Jenvald, J., Morin, M., Timpka, T. & Eriksson, H. (2007). Simulation as decision support in pandemic influenza preparedness and response. In *ISCRAM'07: Proceedings of the 4th International Conference on Information Systems For Crisis Response and Management* pp. 295–304,.

[Koestler, 1967] Koestler, A. (1967). *The ghost in the machine*. Arkana, London, UK.

[Kruchten *et al.*, 2007] Kruchten, P., Woo, C., Monu, K. & Sotoodeh, M. (2007). A human-centered conceptual model of disasters affecting critical infrastructures. In *ISCRAM'07: Proceedings of the 4th International Conference on Information Systems For Crisis Response and Management* pp. 327–344,.

[Lewis & Séquin, 1998] Lewis, R. & Séquin, C. H. (1998). Generation of 3D building models from 2D architectural plans. *Computer-Aided Design,* **30** (10), 765–779.

[Löffler *et al.*, 2007] Löffler, J., Ernst, V. H., Schon, J., Pottebaum, J. & Koch, R. (2007). Intelligent use of geospatial information for emergency operation management. In *ISCRAM'07: Proceedings of the 4th International Conference on Information Systems For Crisis Response and Management* pp. 181–190,.

[Losh, 2007] Losh, E. (2007). The birth of the virtual clinic: game spaces in "the virtual practicum" and the virtual terrorism response academy. In *ISCRAM'07: Proceedings of the 4th International Conference on Information Systems For Crisis Response and Management* pp. 551–556,.

[Luckham & Frasca, 1998] Luckham, D. C. & Frasca, B. (1998). Complex event processing in distributed systems. Technical Report CSL-TR-98-754 Stanford University.

[Miller *et al.*, 2004] Miller, J. A., Baramidze, G. T., Sheth, A. P. & Fishwick, P. A. (2004). Investigating ontologies for simulation modeling. In *ANSS '04: Proceedings of the 37th annual symposium on Simulation* p. 55, IEEE Computer Society, Washington, DC, USA.

[Naylor *et al.*, 1990] Naylor, B., Amanatides, J. & Thibault, W. (1990). Merging BSP trees yields polyhedral set operations. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques* pp. 115–124, ACM Press, New York, NY, USA.

[Ott *et al.*, 2006] Ott, R., Gutiérrez, M., Thalmann, D. & Vexo, F. (2006). Advanced virtual reality technologies for surveillance and security applications. In *VRCIA '06: Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications* pp. 163–170, ACM Press, New York, NY, USA.

[Paoluzzi, 2003] Paoluzzi, A. (2003). *Geometric Programming for Computer Aided Design*. John Wiley & Sons, Chicester, UK.

[Paoluzzi & D'Ambrogio, 1999] Paoluzzi, A. & D'Ambrogio, A. (1999). A programming approach for complex animations. Part II. Reconstruction of a real disaster. *Computer-Aided Design,* **31** (11), 711–732.

[Paoluzzi *et al.*, 1995] Paoluzzi, A., Pascucci, V. & Vicentino, M. (1995). Geometric programming: a programming approach to geometric design. *ACM Trans. Graph.* **14** (3), 266–306.

[Paoluzzi & Scorzelli, 2007] Paoluzzi, A. & Scorzelli, G. (2007). Pattern-driven mapping from architectural plans to solid models of buildings. In *Israel-Italy Bi-National Conference on Shape Modeling and Reasoning for Industrial and Biomedical Applications*, Technion, Haifa, Israel.

[Park & Fishwick, 2005] Park, M. & Fishwick, P. A. (2005). Integrating dynamic and geometry model components through ontology-based inference. *Simulation,* **81** (12), 795–813.

[Pascucci *et al.*, 1995] Pascucci, V., Ferrucci, V. & Paoluzzi, A. (1995). Dimension-independent convex-cell based hpc: representation scheme and implementation issues. In *SMA '95: Proceedings of the third ACM symposium on Solid modeling and applications* pp. 163–174, ACM Press, New York, NY, USA.

[Perrochon *et al.*, 1999] Perrochon, L., Mann, W., Kasriel, S. & Luckham, D. C. (1999). Event mining with event processing networks. In *The Third Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Beijing, China.

[Peterson, 1977] Peterson, J. L. (1977). Petri nets. *ACM Comput. Surv.* **9** (3), 223–252.

[Peterson, 1981] Peterson, J. L. (1981). *Petri Net Theory and the Modeling of Systems*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

[PLaSM Web Site, *a*] PLaSM Web Site. Web pages. http://www.plasm.net.

[Scorzelli *et al.*, 2007] Scorzelli, G., Paoluzzi, A. & Pascucci, V. (2007). Parallel solid modeling using BSP dataflow. *Journal of Computational Geometry and Applications,* **17**. To appear.

[Sebe *et al.*, 2003] Sebe, I. O., Hu, J., You, S. & Neumann, U. (2003). 3d video surveillance with augmented virtual environments. In *IWVS '03: First ACM SIGMM international workshop on Video surveillance* pp. 107–112, ACM Press, New York, NY, USA.

[Smith, 2004] Smith, G. J. D. (2004). Behind the screens: examining constructions of deviance and informal practices among cctv control room operators in the uk. *Surveillance and Society,* **2**, 376–395.

[Snavely *et al.*, 2006] Snavely, N., Seitz, S. M. & Szeliski, R. (2006). Photo tourism: exploring photo collections in 3D. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* p. 42, ACM Press, New York, NY, USA.

[So *et al.*, 1998] So, C., Baciu, G. & Sun, H. (1998). Reconstruction of 3D virtual buildings from 2D architectural floor plans. In *VRST '98: Proceedings of the ACM symposium on Virtual reality software and technology* pp. 17–23, ACM Press, New York, NY, USA.

[Teller, 2003] Teller, S. MIT building model generation (BMG) project. http://city.csail.mit.edu/bmg/.

[Williams *et al.*, 2006] Williams, S., Shalf, J., Oliker, L., Kamil, S., Husbands, P. & Yelick, K. (2006). The potential of the cell processor for scientific computing. In *CF '06: Proceedings of the 3rd conference on Computing frontiers* pp. 9–20, ACM Press, New York, NY, USA.

[Wu *et al.*, 2006] Wu, E., Diao, Y. & Rizvi, S. (2006). High-performance complex event processing over streams. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data* pp. 407–418, ACM Press, New York, NY, USA.