# An integrated system for disaster management in industrial areas

P. Arena[1], L. Patanè[1], S. Caruso[1], M. Anastasi[1] & A. Cannata[2]
[1]*DIEES- University of Catania, Italy*
[2]*Euroconsult s.a.s., Italy*

## Abstract

In this paper an integrated architecture, named Juan Chedan, for disaster management is proposed. The modular structure of the framework includes a core that manages streams of data from different plugins used for: data acquisition, accident simulation and result reporting. Moreover the integrated system allows geo-localization and data logging from remote devices, such as sensors, distributed in industrial areas and equipped on mobile robots.

*Keywords: emergency management, data gathering, robot controlling.*

## 1 Introduction

The study of the potential environmental impact due to an industrial incident through predictive models is an integral part of risk analysis, but is probably not accurate enough to guarantee a correct response to real incidental scenarios, and a correct estimation of the actual risks and potential consequences of a particular event.

It's possible to act in multiple ways to achieve an improvement of this scenario: one solution consists into speed up the assessment operations by integrating several systems needed to collect actual data from the environmental context, e.g. temperatures, pressure levels, humidity, wind speed and direction, etc.

The framework discussed in this work allows this integration both at the beginning of the control process during data collection, and in progress for online control, in order to collect useful information for the validation and correction of the prediction algorithm.

WIT Transactions on The Built Environment, Vol 110, © 2009 WIT Press
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/DMAN090331

The data acquisition is enabled through the use of robot probes containing on board sensors that are able to explore dangerous areas, retrieve samples and perform online measurements in order to give a feedback to the control framework. The framework can also gather data coming from probes or fixed sampling stations, generate alarms and route them towards external central emergency management servers.

All the online and offline data gathering, robot probe controlling and emergency management functionalities are integrated in the main framework through the use of separate software plugins. The main framework's role is to offer a common visualisation and interaction interface based on geo-referenced maps to enable real time tracking of data collection and monitoring of the robot probes.

## 2 The architecture

The Juan Chedan platform is characterized by a modular structure where the application Core (responsible for input, data gathering, data presentation), is completely separated from the models, the remote tracking functions and even the output visualisation as shown in Fig. 1.

The most important feature of the framework is its modular structure, which potentially makes the software dynamically adaptable to any model, unlike many similar prediction suites that are specifically built around a model or family of models [1].

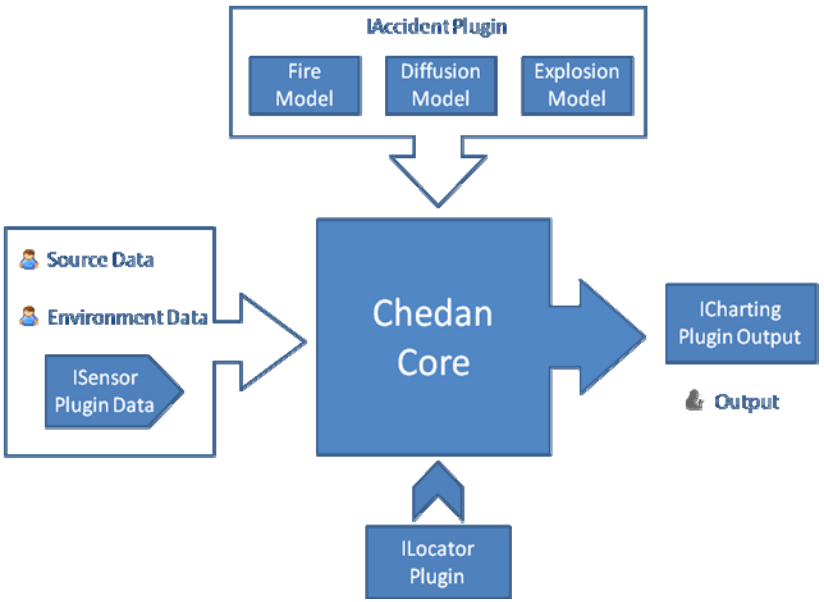


Figure 1: Juan Chedan architecture.

### 2.1 The Juan Chedan Core

The Juan Chedan Core system manages the stream of data between the different sub-systems shown in Fig. 1. It uses data describing the accidental source and the environment, which may be provided manually by the user or automatically using sensors.

The Core uses the mathematical models implemented in separate Plugins to process the input data and produce output data that can be overlaid on a map. The output can be shown as a vulnerability map overlapping a digital vector map or a photogrammetric map. The output data collections generated by the plugins can also be presented as two-dimensional charts, which is especially useful when presenting time-dependent trends. To provide this kind of functionalities, the role of each plugin is to implement a specific interface (IChartingPlugin). The Framework's capabilities may also be extended for further cases of study, concerning geo-localization and logging of data from remote devices (e.g. for robot tracking) or from sensors, cameras and other mobile devices.

#### 2.1.1 The Juan Chedan GUI

A simple user interaction with the application is guaranteed by the use of a Graphical User Interface, presenting a set of standard features that are familiar to most average computer users. It shows a very simple and intuitive environment for emergencies management. The interface is composed of 4 parts as shown in Fig. 2:

- Workspace
- Project Explorer
- Ribbon Bar for data input
- Logging area

*The Workspace* is the main part of the interface. The Microsoft® Virtual Earth™ maps are shown here, together with the overlaid scenario information [2]. The visualised information consists of:

- Risk sources;
- Vulnerability areas (areas to be supervised);
- Observer points (points where risk level trends will be computed);
- Dynamic vulnerability maps;
- Tracking points transmitted by mobile agents (remotely driven robots, etc).

*The Project Explorer* is the other main part of the user interface. It presents the user with a structural view of all the elements involved in the current simulation scenario. All the entities are grouped in a tree view. The root represents the current project, and may contain four kinds of nodes:

- Risk sources;
- Observation points;
- Generated vulnerability maps;
- Geographic data layers.

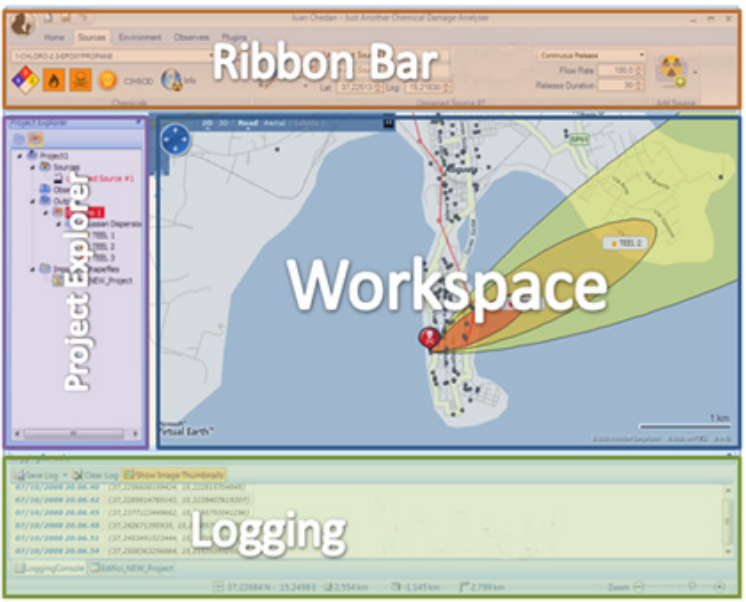


Figure 2: Graphical Interface of Juan Chedan.

## 2.2 Plugins

The Juan Chedan Plugins are stand-alone software components that implement specific models offering their functionalities to the host Core. Usually the plugins don't have a graphic interface and interact with the user only through the Core [3, 4].

Different plugins may offer a wide range of different functionalities and behaviours and their nature is defined according to the programming interface they implement. A list of the currently developed plugin is given in the following together with a brief description.

### 2.2.1 Model plugins (IAccidentPlugin)

These plugins implement mathematical models for risk scenario simulation. Currently, three different types of scenario are considered:

- Toxic clouds dispersions
- Explosions
- Fires

For more details about the accident plugins see [1].

### 2.2.2 Localization and tracking plugins (ILocatorPlugin)

Juan Chedan is able to track and visualise data originating from remote devices. By implementing the ILocatorPlugin interface it is possible to connect to a remote data source, visualise its output data and track its position on a map using any plugin-defined communication channel.

The plugins that have already been implemented and are currently available are:

• WMDeviceTrackerAsmxPlugin and WMDeviceTrackerSocketPlugin, which allow remote tracking of Microsoft Windows Mobile-basedagents, and allow to retrieve data from their onboard sensors, using either a SOAP Web Service or TCP/IP as a communication channel.

• FileWatcherTrackerAsmxPlugin, which offers remote tracking and sensor data retrieval functionalities to any device capable of saving geo-referenced data on files in a textual format (Csv, Fixed-Width…). The communication between the remote device and the plugin is again achieved using a SOAP Web Service. An example of remote traking of a mobile robot is shown in Fig. 3.

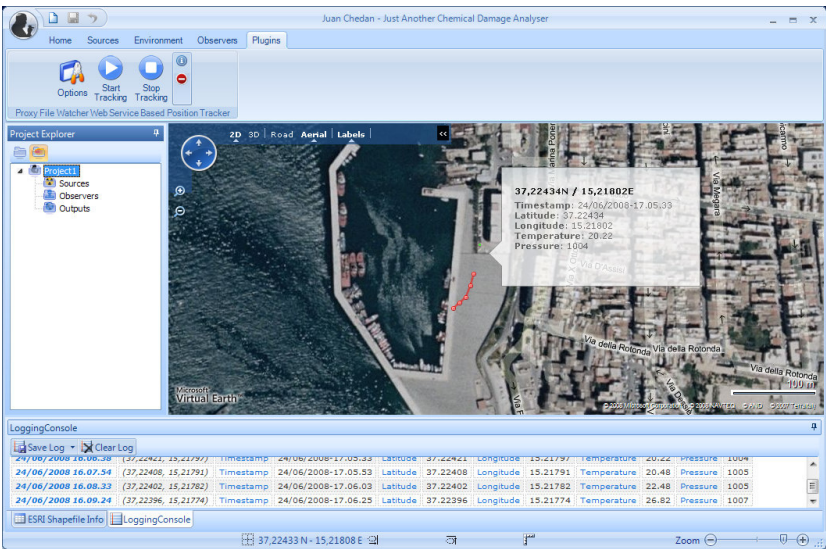


Figure 3: Remote device tracing.

### 2.2.3 Output and charting plugins (IChartingPlugin)

The expandable architecture of Juan Chedan keeps the Presentation Layer completely separated from the application core to improve the flexibility.

The same "generic" approach used by the Core for gathering the processed output from the plugins, is used for sending output to the visualisation plugins, once more using a specific communication interface.

It's thus possible to use any visualization component, written in any language and even developed from third parts, as long as it is wrapped in a .NET plugin that implements the simple interface requested by the Core in order to identify the component as a data visualisation tool and establish a communication with it. (Fig. 4).

At the present time the implemented charting plugins are:

- Amline Chart (leveraging a commercial charting library using html and Macromedia Flash) [5].

- ZedGraph Chart (leveraging an open source graphic library written in C#) [6].

### 2.2.4 Input plugins (ISensorPlugin)

Input plugins may be used to provide the Core with data retrieved from sensors, web services, files, etc. These data are presented to the core in a common format so that they can be treated uniformly, independently of their origin. It is thus possible to avoid manual input of environmental data (temperature, relative humidity, wind direction and speed, etc.).

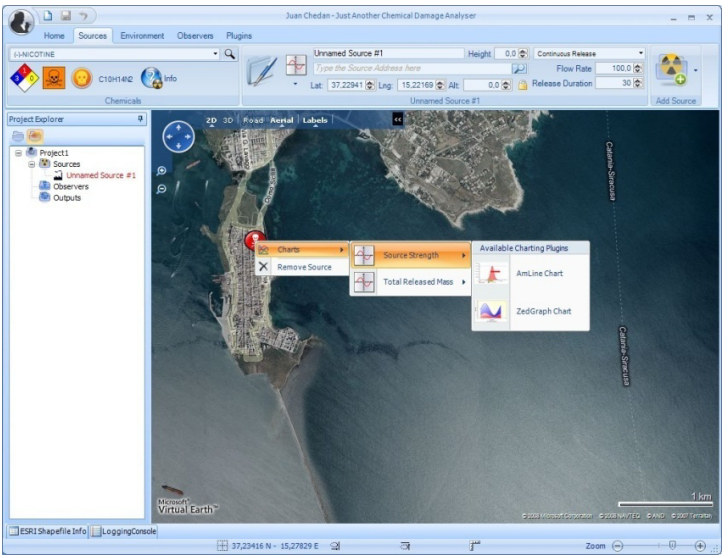


Figure 4: Run time visualisation of the available charting plugins.

### 2.2.5 Sink plugins (ISinkPlugin)

The architecture includes the functionality to redirectdata to a specific remote entity in real time. For example, this capability was used to redirect remote device tracking data towards a control centre that supervised, observed and coordinated the emergency response operations in order to generate alarms if necessary.

To fulfil this need, we introduced a new category of plugins to be used as information "sinks". These plugins enable the Core to send tracking data to any subscribed interested party; once a sink plugin starts to receive information, it can process it independently, and perform any operation necessary to use it appropriately (generate a report, connect to a remote service and relay the data).

## 3 Communication suite

A complete suite of distributed applications was developed during the course of the creation of Juan Chedan. These applications communicate with each other

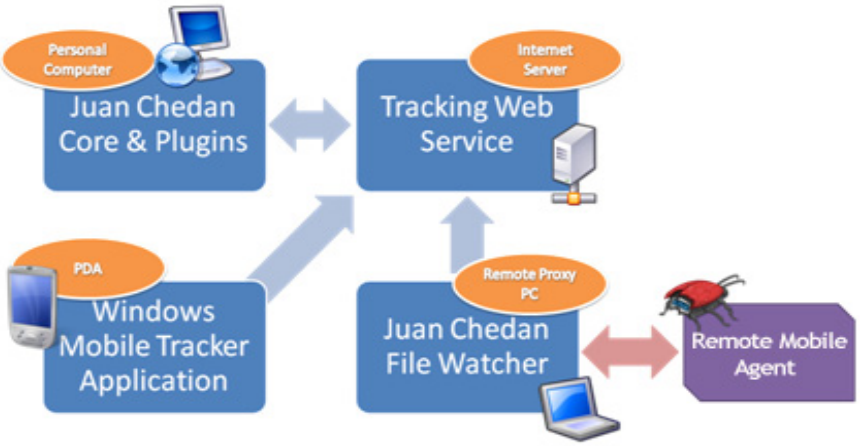


Figure 5: Deployment diagram.

using multiple strategies and protocols. A scheme of the whole structure, including the external applications is shown in Fig.4. A description of each application is given in the next sections.

### 3.1 Juan Chedan File Watcher

Juan Chedan File Watcher is an utility that was developed to allow the interaction of the Juan Chedan framework with a legacy application for remote agent control.

In the original architecture, the remote mobile agent communicates with a laptop situated at a short distance using radio devices. The data retrieved from the robot is dumped in real time to a text file.

Juan Chedan File Watcher keeps watching this dump file and whenever new data is available, it elaborates the information, applying filters and transformations, and then relays it to a Juan Chedan input plugin, basically acting as an intelligent proxy between the robot and the control station where Juan Chedan is running.

This architecture allows any application able to dump geo-referenced data to a text file to communicate with Juan Chedan and integrate with its framework, effectively becoming part of its architecture. Thus, any data source may potentially be integrated in Juan Chedan and manipulated with simplicity, greatly improving the user experience thanks to the unification of multiple heterogeneous data sources in a single environment. A complete scheme of the File Watcher interface is given in Fig. 6.

### 3.2 The tracking web service

In order to avoid firewall-related communication problems and make the tracking functionality more simply and readily usable even for a low-privilege

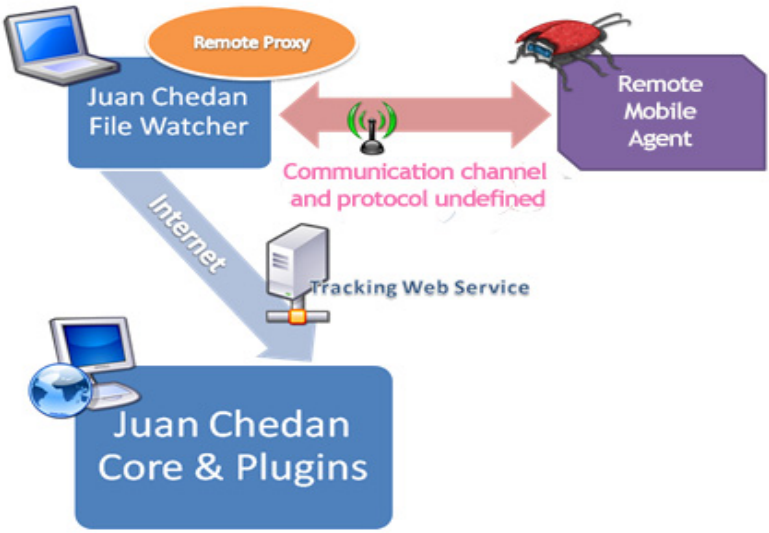


Figure 6: Juan Chedan File Watcher: communication by proxy.

computer operator without administration rights on his/her local network, the communication between the proxy described in section 3.1 and Juan Chedan is not based on raw TCP/IP, but uses a SOAP Web Service installed on a remote Internet server.

This approach allows both the subjects involved in the communication (the remote proxy and the workstation where Juan Chedan runs) to avoid the use of not well-known TCP ports, which might be blocked by a firewall. The only port used for communication with the Tracking Web Service will be the standard HTTP port (TCP port 80), that is usually open on most computers.

### 3.3 Windows Mobile Tracker

Windows Mobile Tracker is an application for Microsoft® Windows Mobile 2003 SE PDAs or Smartphones having a GPS (Global Positioning System) module and a wireless Internet connection [7].

This application retrieves the geographic position of the device using the on-board GPS unit and communicates it to a tracking plugin that runs within the Juan Chedan environment on a remote workstation. The communication is carried out through the wireless Internet connection and uses the Web Service described in section 3.1. In addition to the device position, the application can transmit any kind of custom data received from sensors connected to the Windows Mobile device. Fig. 7 shows the communication flow used to transfer the data to the Juan Chedan Core.

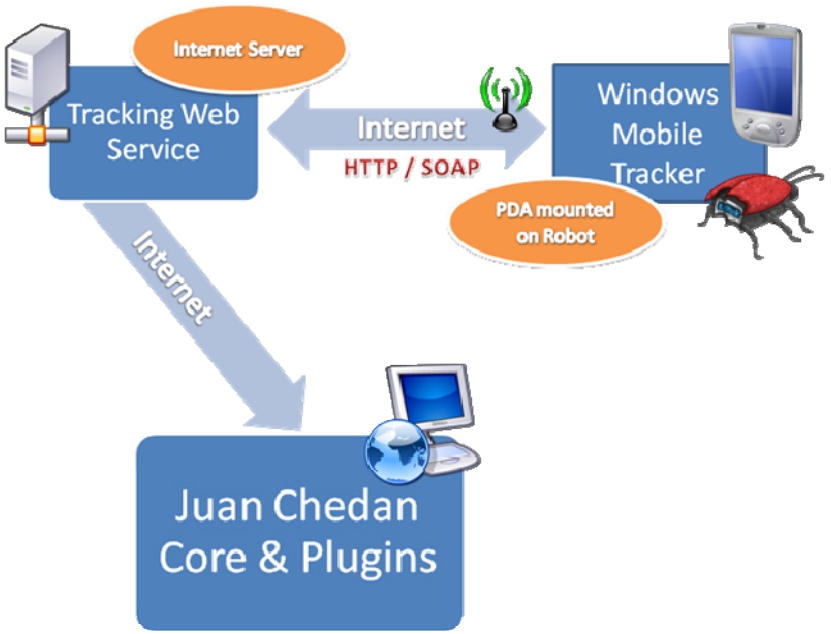


Figure 7: Communication between Windows Mobile Tracker and Juan Chedan.

## 4 Conclusions

In conclusion we can assert that although the prediction of the potential environmental impact consequent to an industrial incident is often considered the most important aspect of risk analysis, the core ability of a software designed to help responders to deal with emergencies should be to help the operators to easily integrate data originating from multiple sources as readily as possible and thus enable them to evaluate and communicate the potential risks and consequences of an event.

Obviously, where the prediction algorithms are the brain, robots carrying on-board sensors, able to explore incidental areas, take samples and measurements, are the hands.

The framework can elaborate the gathered data using a set of different algorithms and appropriately originate alarms and route them towards external emergency management centres.

All the functionalities related to online and offline data gathering, robot probes control and emergency management are integrated with the main framework through the use of separate software plugins in order to maximise flexibility and expandability.

## Acknowledgement

This work was partially supported by the Isemiha project, (1999.IT.16.1.PO.011/3.14/5.2.13/0312).

## References

[1] Arena, P., Patanè, L., Anastasi, M., Caruso, S. & Cannata, A., *A software framework for the generation of dynamic vulnerability maps for riskassessment*, Int. Conf. On Disaster Management 2009, UK.
[2] Microsoft Developer Network, *Virtual Earth Development Centre*. Available on line at: http://msdn.microsoft.com/en-us/virtualearth/default.aspx
[3] Löwy, L. *Programming .NET Components*, O'Reilly 2003, USA
[4] Szyperski, C.: *Component Software: Beyond Object-Oriented, Programming. 2nd ed.* Addison-Wesley Professional, Boston 2002
[5] *AmCharts Documentation*, 2008. Available on line at: http://www.amcharts.com/docs/
[6] *ZedGraph Wiki*, 2007. Available on line at: http://zedgraph.org/wiki
[7] Baddeley, G. - *GPS - NMEA sentence information* 2007. Available on line at: http://home.mira.net/~gnb/gps/nmea.html