DISTRIBUTED EMERGENCY MANAGEMENT WITH SPATIAL SCENARIOS

Peter Sapaty
Institute of Mathematical Machines & Systems, National Academy of Sciences,
Glushkova Ave 42, Kiev 03187, Ukraine

Robert Finkelstein
Robotic Technology Inc., 11424 Palatine Drive, Potomac, Maryland 20854, USA

Joaquim Filipe
Departamento Sistemas e Informática, Escola Superior de Tecnologia de Setúbal,
Setúbal 2910-761, Portugal

Keywords: Critical infrastructures, key resources, emergency management, emergent societies, crisis relief, information technologies, distributed control, WAVE-WP model, wearable electronic devices, mobile robotics, spatial scenarios.

Abstract: A radically new approach will be described for the fully distributed and dynamic management of advanced crisis relief operations and missions. It is based on the installation of a universal "social" module in many existing and massively used data processing and control devices, including (but not limited to) internet hosts, laptops, mobile robots and mobile phones. These modules can collectively interpret a special scenario language while exchanging higher-level program code with accompanying data and control in parallel. This can dynamically integrate any scattered post-disaster human and technical resources into an operable distributed system which, from one side, is effectively supervised externally, and from the other side, is capable of solving complex self-analysis, coordination, survivability, relief, and reconstruction problems autonomously.

1 INTRODUCTION

1.1 The Grim Big Picture

Millions of people are on the move, traffic jams everywhere. Houses destroyed, infrastructures gone, winds hundreds kilometers per hour, flooding and fires. No electricity, shortage of food and fuel, usual ties broken, businesses vanished, jobs lost. No central authorities or services, looting and lawlessness… This is becoming a familiar picture throughout the world, especially due to global warming and climate change. Katrina and Rita are the recent sad examples. Earthquakes are another disaster area, like the recent one in Pakistan, and the tsunami a year before. Manmade disasters caused by armed conflicts and terrorist attacks are effectively contributing to this list too.

How to regain integrity, restore law and order, and assemble scattered resources for a collective survival? How to rebuild the damaged territory, revive the previous infrastructures or create new ones, and return to normal life?

In Fig.1, a symbolic picture of the post-disaster area, once representing an integral organism, is shown with the wreckage of living quarters, organizations and infrastructures, also separated and scattered individuals and their emergent grouping.

Despite indiscriminate damages, the disaster area can still hold key human, technical and natural resources; its parts can still be able to communicate with each other. For example, cellular towers, at least some of them, can still be operating, and/or

Sapaty P., Finkelstein R. and Filipe J. (2006).
DISTRIBUTED EMERGENCY MANAGEMENT WITH SPATIAL SCENARIOS.
In *Proceedings of the Third International Conference on Informatics in Control, Automation and Robotics*, pages 57-66
DOI: 10.5220/0001221500570066

access to internet may remain available. Radio communications, usually local, can be helpful too, and some units in the area may have satellite phones. The electric grid power may remain available, and the same grid can potentially be used for data communications too (such promising projects already exist). Possible stratospheric solutions for broadband communications, which are discussed now too, may look promising to cope with the crisis situations. So there may be sufficient resources for self-survival and even self-recovery after the major disasters, especially with an external aid hurrying to the area, but they may be highly scattered and very much disorganized.

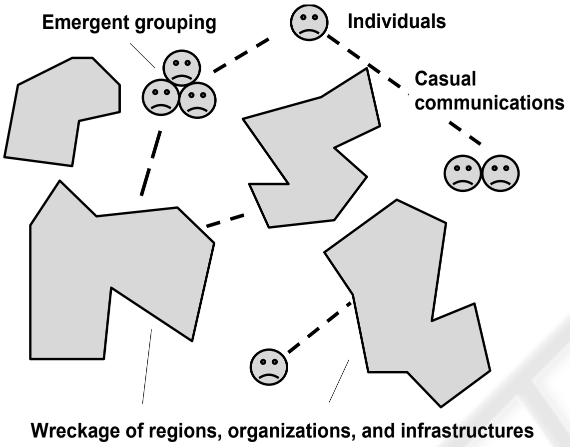

Figure 1: A grim picture of the disaster area.

Restoring *integrity and coordination* in the disaster area is becoming a primary goal in emergency management efforts.

1.2 Emergency Management

Emergency management, EM (Bullock, 2004), due to the increased world dynamics, is one of the hottest topics today. The emergency managers around the world are faced with new threats, new responsibilities, and new opportunities. It is widely believed (Nunn, 2005) that the combination of novel technologies and data bases can allow law enforcement and intelligence investigators to identify potential terrorist plots, use a multitude of data bases that may contain hidden patterns of information about transactions needed to execute plots, and then mount preemptive strikes to stop their plans.

The National Response Plan (2004) establishes a comprehensive all-hazards approach to enhance the ability to manage domestic incidents. The plan incorporates best practices and procedures from different incident management disciplines. Another prominent document, The National Infrastructure Protection Plan, NIPP (2005) provides a unifying structure for the integration of all critical infrastructures and key resources protection efforts into a single national program.

However, the existing efforts are actually offering yet another infrastructures, to be built on the same principles as the existing ones, i.e. consisting of specialized components located in certain places and communicating with each other, with a good deal of central control over them. Due to this, they may inevitably be as (if not more) vulnerable to attacks and failures as other infrastructures, and may become a burden rather than savior.

Even in the relatively modest Katrina case, local infrastructures were indiscriminately fragmented and totally inoperable (while malicious ones thriving), and federal bodies showed clumsiness and inefficiency. The new global infrastructures outlined, like what we see in NIPP, may result in a similar performance in case of major disasters caused by hurricanes, earthquakes, or WMD attacks.

1.3 New Approaches Needed

We believe that the critical infrastructure protection, recovery, and relief ideologies and technologies should be based on quite different, revolutionary rather than evolutionary, principles, and they should evolve and operate in other time-space dimensions than the traditional infrastructures and forces that can harm them, in order to be incomparably superior and unaffected themselves in case of major crises.

A completely different approach is being developed of how to penetrate into any distributed and open systems and establish an overwhelming power over them (destruction of malicious infrastructures in these systems being an option).

Any global or local scenario we want to implement over any area or system is formulated in a special spatial control language, which is interpreted cooperatively in a distributed system widely using smart mobile program code covering the system or its parts in parallel. This provides spatial hologram-like algorithms which dynamically exist in between system components rather than inside them, often being unobservable and unreachable by conventional means.

Actually, this work follows a sort of an intelligent super-virus ideology, with its potentially unlimited self-penetration and self-recovery possibilities. The approach can also setup, at runtime, any needed infrastructures over scattered post-disaster human or technical resources, and these infrastructures can evolve and freely migrate in both physical and virtual worlds, self-recovering after damages and preserving integrity and goal orientation. The paper summarizes the technology called WAVE-WP that serves these purposes, outlines its applications and possible implementation with the use of massively wearable devices. Examples of spatial programming of some exemplary relief operations will be demonstrated too. The paper also uses and extends the material previously reported at (Sapaty, Sugisaka, 2006).

2 THE WAVE-WP PARADIGM

The distributed computation and control WAVE-WP (or World Processing) model and technology (Sapaty, 1999, 2005; Sapaty, Sugisaka, 2005) are based on a higher-level language describing parallel distributed solutions in computer networks as a single seamless spatial process rather than traditional collection and interaction of parts (agents), while shifting these and other routines to an efficient automatic implementation.

Communicating copies of the WAVE-WP language interpreter (WI) should be present in sensitive points of the system to be governed (like internet hosts, robots, troop carriers, dismounted soldiers, separated groups of individuals, laptops, mobile phones, etc.), as shown in Fig. 2 for the post-disaster area of Fig. 1.

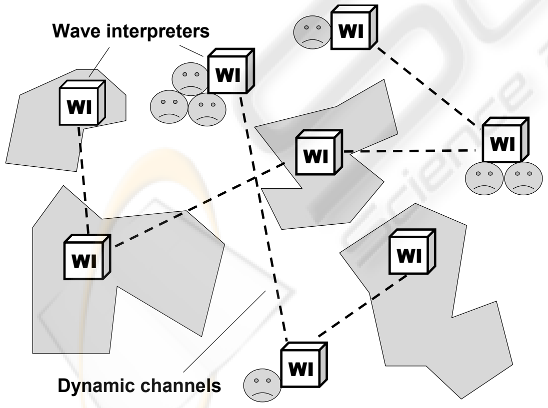

Figure 2: Wave interpreters in sensitive points of the distributed system.

Parallel spatial scenarios (or waves) written in the language can start from any interpreter, covering the network at runtime and cooperating with each other in the distributed space. The approach often provides hundreds of times application code reduction and simplification, allowing us to concentrate on efficient global solutions rather than implementation details.

Spreading via networked WIs, waves can create dynamic knowledge infrastructures arbitrarily distributed between the system components. Subsequently or simultaneously navigated by same or other waves, they can effectively support distributed databases, advanced command and control, global situation awareness, parallel inference, and autonomous decisions. It is convenient to operate in this seamless virtual world fully ignoring its physical distribution, whereas virtual networks can migrate (partially or as a whole) in physical networks while being processed.

Installed in advance (or loaded in an emergency) in different components that may happen to be located in a disaster area, with possibilities of their communication using any remaining channels, WIs can convert the whole area into an operable self-organized system. The latter being able to solve complex tasks on itself, where high-level relief scenarios in WAVE-WP can start and evolve from any interpreter, covering the whole system or its needed parts at runtime (see Fig.3).

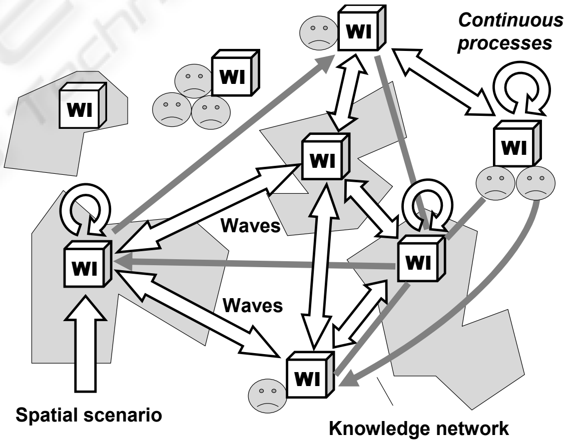

Figure 3: Converting the disaster area into an operable system.

The technology has been working successfully in (mainly static) computer networks since 1990, with many computers distributed between different countries, via the Internet. From 1993, many complex parallel and distributed solutions were shown with the creation of distributed virtual infrastructures mounted between UK, Germany, Canada, and the US, also multiple chases of mobile agents & objects by the other ones throughout the globe. Information about these projects can be found in (Sapaty, 1999).

The current orientation of this activity has been fully on highly dynamic networks, which may include Internet, mobile cellular networks, radio and satellite communications, etc. There may also be no networks at all in the post-disaster zone, in the usual sense. In the worst case, people can see each other at a distance (e.g. sitting on floating roofs and shouting) passing manually the higher-level commands and accompanying data in the user-friendly WAVE-WP version. The people can happen to be on a constant move, changing casual links with each other over time.

The technology is fully orienting on such loosely connected, dynamic and emergent systems. We can write arbitrary complex distributed algorithms finding the needed solutions in completely distributed environments with such unsafe and emergent links. This can allow us to start both local and global businesses in the disaster zone without restoration of regular communications, which, of course, would be much helpful.

3 WAVE-WP LANGUAGE

3.1 The Language Basics

We are considering here only the top definition of the WAVE-WP language, shown in Fig. 4, with more details in (Sapaty, 1999, 2005).

wave → { *advance* ; }
advance → { *move* , }
move → *constant* | *variable* | [*rule*] (*wave*) |
{ *move act* }
rule → *forward* | *echo*
act → *fusion* | *flow*
variable → *nodal* | *frontal* | *environmental*
constant → *information* | *matter* | *code*

Figure 4: WAVE-WP language syntax.

Starting from a certain position, the program, or *wave*, navigates in physical or virtual space, with successive *advances* (separated by a *semicolon*) starting from positions reached by the previous advances. An advance may consist of *moves* (separated by a *comma*) which can develop in parallel, each one from the same position.

Moves may: reflect the result directly, as a constant or variable; represent any wave in parentheses optionally prefixed by a rule; be arbitrary expressions where other moves are separated by elementary operations, or acts.

Rules being *forward* rules, coordinating spreading of waves or setting up special navigation contexts, or *echo* rules detailing the fusion and return of (remote) states and data. *Acts* classify as *fusion acts* producing new values from operands, and *flow acts* moving data and control in space. The same operations on waves may be set up by both rules and equivalent to them acts within the expressions. *Variables* may be: *nodal*, dynamically associated with space positions and shared by waves; *frontal*, moving in space with control; and *environmental*, accessing the navigated environment in points reached.

Constants and variables may represent both *information* and physical *matter*; they may also represent program *code* to be created, processed or modified with a subsequent execution as waves, thus providing programming flexibility in dynamic environments.

3.2 Spatial Interpretation

What follows from this language definition, is the unwrapping and replication of the recursive formulae, rather than traditional reduction, as shown in Fig. 5.

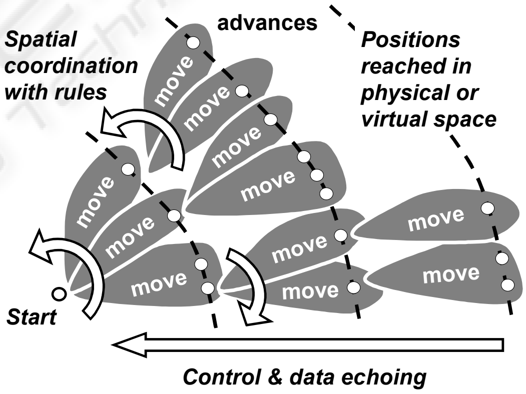

Figure 5: Spatial evolution of waves.

The wave program matches, conquers, floods, covers the distributed physical or virtual world in parallel, establishing full control over the space. Each act is performed in the reached space positions on local data there, or on what is obtained and returned by other waves of the expression. The resultant value on any construct comprises all values obtained in the points of space reached by it. All constructs return control states, which are merged and generalized on higher levels with the use of rules, for making hierarchical spatial decisions.

A number of successful implementations of this approach have been made in different countries, with public domain of the previous implementation available on the Internet, for example, in Canada (Wave system, 2003). The new, advanced, version is currently being re-implemented and patented, with orientation on both software product and direct “wave chip”.

4 USING WEARABLE DEVICES

Emergency management may be fundamentally assisted by the massively wearable individual data processing and communication devices, which are expected to remain with individuals under any circumstances. WIs, installed in them (this can be done without problems, taking into account the existing experience of implementation of the WAVE system on different platforms and compactness of the language interpreter) can make the whole societies, especially emergent ones in the disaster areas, to be programmable and controllable in the way required, despite the scattering of resources and limited communications.

Mobile phones, the undisputed leader among wearables, are expected to be the most common consumer electronics device on the planet (Mobiles, 2005). By the end of 2009, some 2.6 billion mobiles will be in regular use around the world. Mobiles, for example, are catapulting rural Africa into the 21st century, making it the world's fastest-growing cellphone market (LaFraniere, 2005). They are enabling millions of people to skip a technological generation and bound straight from letter-writing to instant messaging. Asia is the next fastest-expanding market.

Many also use GPS, digital cameras, portable PCs along with mobile phones, and there already exist advanced products that combine all these and many other features within a single piece (including also Bluetooth, WiFi, phone radio and TV (GPS, 2006; Samsung, 2005). 3G, or third generation, brings broadband for mobiles too (3G, 2005). Also, the recently unveiled sub-$100 laptop design (Sub-$100, 2005), with its internet wireless access and a hand crank for when there is no power supply, is expected to contribute drastically to the unprecedented electronic integration of the whole world. Some of the currently popular massively wearable devices are shown in Fig.6.

We may assume that these devices (at least some of them) would remain able to communicate with each other during and after the disasters via the wireless networks, internet, radio or any other channels, and in the worst case even via humans (who, using voice, handwriting or gestures, can pass manually the needed code and data, with the return to an electronic WI level at the reception end).

Figure 6: Examples of massively wearable devices.

5 MANAGEMENT EXAMPLES

We will consider here only elementary examples of spatial programming in WAVE-WP for some tasks that may relate to emergency management, with code simplifications necessary to fit the limited paper space.

5.1 Spatial Counting of Casualties

Let us consider a fully distributed and parallel counting of a total number of casualties in the disaster area, on all affected regions, assuming for simplicity that only a single WI is used for this task in each region or group of individuals. (The interpreter-participant can be negotiated locally if more than one available in the group/region.)

The following program can be applied from any WI as an entry or starting one, which can be far away from the disaster area (say, located in a federal center). This entry node can also happen to reside within the area as a special or emergently selected body or individual (or one of them).

```
frontal(Domain);
Domain = <disaster area>;
USER =
 sum(
  hop(direct, Domain);
  repeat(
   done(apply(casualties)),
   hop(anylinks, Domain, first)))
```

Let us explain its work step by step, where various parts of the program can replicate and operate in different places of the distributed word, in parallel, always preserving scenario integrity and subordination to global control. The latter being parallel and distributed too.

```
frontal(Domain)
```

This declares variable `Domain` in the starting node as a frontal one, which will subsequently propagate with the program control in a distributed space, as its property.

```
Domain = <disaster area>
```

In the same node, `Domain` is assigned proper parameters of the disaster area (represented, say, as a polygon or a set of them, here not detailed further), sufficient to determine whether a point with given coordinates may belong to the area of concern.

```
USER = sum( … )
```

Initiates in the starting node a distributed process (enclosed in braces) covering the whole disaster area and finding casualty numbers in different places (regions) in parallel, summing them in parallel too. The final result will be displayed to the human operator in the entry node (represented by environmental variable `USER`). The top control of all these (hierarchical) processes will remain in the starting node.

```
hop(direct, Domain)
```

This tries to make a direct, parallel, electronic hop from the starting node to as many directly reachable units within the area in `Domain` as possible. The hop may use for this any available means and channels for finding such nodes (existing records, databases, and cable, cellular, satellite or radio links, with selective or broadcasting mode of operation). The number of nodes accessed directly may be limited, especially if the communication infrastructures in the area are damaged, and also it is far from the starting node.

```
repeat( … )
```

From the node(s) reached above, starts a spatial navigation loop throughout the whole disaster area, with the following two parallel branches in its body.

```
done(apply(casualties))
```

This first branch in each reached node activates an external procedure `casualties`, which counts the number of humans badly affected in the current region, resulting in forming of an open value of this branch. Enclosure of the branch by rule `done` makes it also a terminal one in each node (i.e. from which the spatial loop will not continue). The `casualties` procedure may be fully automatic, performed, say, by special observation robots; automated, with manual use of electronic equipment to check damages; or fully manual, with casualties counted by people visually and then returned to the electronic level via a terminal to the wave interpreter.

```
hop(anylinks, Domain, first)
```

This second branch of the loop tries to hop from a node in the disaster area to other nodes of the same area, given in `Domain`. It uses any available local communication means from the current node which, for example, may be registered in its mobile phone as dial numbers of people this person knows and communicates regularly. These may also be still operating radio channels, usually of a limited distance, or longer range Bluetooth-like direct wireless communications used between different mobile phones and PCs (i.e. not needing cellular towers or Internet that may be damaged). And finally, in the worst case, it may just be passing user-friendly WAVE-WP commands and data by voice or handwriting to other persons in direct contacts.

This hop assumes contacting as many neighboring nodes and as simultaneously as possible. To exclude duplications of actions on the same request stemming from different neighbors, the nodes will react only if the request comes to them first time (additional parameter `first` in the hop). In all the new neighbors reached by this parallel hop, the full body of the loop, i.e.:

```
repeat( … )
```

will start again, with the terminal `done` branch assessing casualties, and the hopping branch trying to reach its own neighbors within the area, prolonging the loop until there are neighbors visited first. The terminal branches will leave casualty numbers in all nodes reached in the area, which will subsequently be summed up, returned, and finally output to the `USER`.

On the internal implementation layer, invisible for a user, this spatial repetitive parallel program forms a runtime spanning tree covering (if communications permit) the whole disaster area via the embedded and reached wave interpreters in it, with the discovered local casualty numbers suspended in all the tree nodes. These numbers are subsequently collected and summed up in parallel when echoing from the fringe nodes and up the tree, finally receiving the total number of casualties at the root node. As can be seen, the whole disaster area has been effectively converted into a runtime parallel spatial (here tree-structured) machine, capable of solving the problem in a fully distributed manner and without any central computational resources, as shown in Fig. 7.

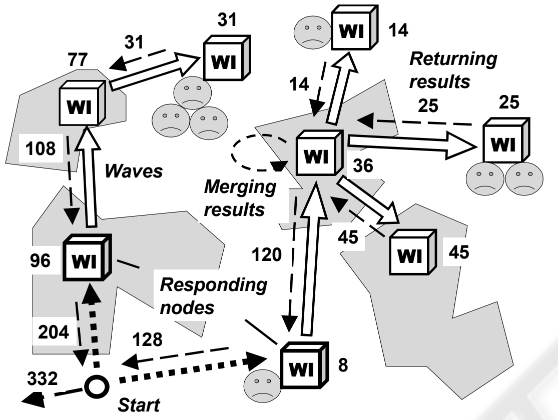

Figure 7: Spatial counting of the total number of casualties.

5.2 Delivery to an Affected Region

If to use echo rule `max` instead of `sum` of the previous program, and also lift physical coordinates of the navigated regions, it is possible to get the final result on the disaster area as an aggregated maximum casualties-location value, as follows:

```
frontal(Domain); nodal(Max);
Domain = <disaster area>;
Max =
 max(
  hop(direct, Domain, random);
  repeat(
   done(apply(casualties)_WHERE),
    hop(anylinks, Domain, first)))
```

The first part of the result in nodal variable `Max` at the entry node will give the number of casualties of the most affected region, and the second part will provide physical coordinates of this region. The latter -- as measured from physical position of the related WI by using a special environmental variable `WHERE` associated, say, with GPS devices.

The underscore here represents an aggregation act, with the first part of the compound values obtained in all regions behaving as keys for the spatial rule `max` (operating in parallel on the runtime spanning tree of the whole area). The second part serves just as an attachment to this key.

Using the obtained value in `Max`, it is easy to describe in WAVE-WP an assemblage of the needed number of relief packages for the most affected region (as objects `“relief”`, with physical matter in the language identified by double quotes). These objects can be multiplied on the number of casualties (as the first part in `Max`, using the colon as an indexing act), with assignment of the physical result to frontal variable `Supply`, as follows.

```
frontal(Supply);
Supply = “relief” * Max:1;
hop(direct, Max:2);
apply(distribute, Supply)
```

The rest of the program provides direct movement into a physical location identified by the second part in `Max`, and distribution, upon reaching the destination, of the relief packages between individuals that may need them, using external (manned, or robotic) procedure `distribute`.

5.3 The Delivery to All Regions

It is easy to modify the previous two programs in order to find casualty numbers separately on all affected regions, with corresponding coordinates of these regions, and then pack the needed amount of goods for every region and forward all these to the proper destinations, with a subsequent distribution.

The collection of casualty numbers on all regions of the disaster area will be as follows:

```
frontal(Domain); Nodal(All);
Domain = <disaster area>;
All =
 collect(
  hop(direct, Domain, random);
  repeat(
   done(apply(casualties)_WHERE),
   hop(anylinks, Domain, first)))
```

After the spatial work of this program, in a nodal variable `All` at the entry node will be the list of aggregated two-point values on all regions reached in the parallel distributed loop. The following program splits the list on its aggregated scalar values to be used in parallel by the rest of the program.

```
frontal(Supply);
split(All);
Supply = “relief” * VALUE:1;
hop(direct, VALUE:2);
apply(distribute, Supply)
```

After splitting, the rest of the program will replicate on the implementation layer into identical branches with different starting open values (the latter accessed by an environmental variable `VALUE` individual for each branch). Each branch prepares its own physical collection of relief packages for the corresponding region (using first part of `VALUE`), which will be delivered by individual coordinates for this branch & region (second part of `VALUE`).

The above two programs can be effectively integrated into a single spatial program, as follows:

```
frontal(Domain, Supply);
Domain = <disaster area>;
split(
 collect(
  hop(direct, Domain, random);
  repeat(
   done(apply(casualties)_WHERE),
   hop(anylinks, Domain, first)
  )));
Supply = “relief” * VALUE:1;
hop(direct, VALUE:2);
apply(distribute, Supply)
```

The needed number of packages will be delivered to the related destinations via routes available, as shown in Fig. 8.

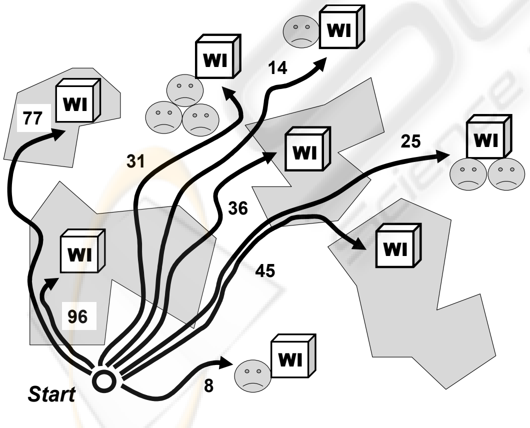

Figure 8: Delivery of the goods to affected regions.

This may take into account the remaining usable road infrastructures, also peculiarities of the terrain. This delivery may be performed by manned vehicles or convoys, or by advanced unmanned mobile ground systems like those described in (Joint robotics, 2005). Any concrete delivery, including path finding and avoidance of obstacles, can be effectively represented in WAVE-WP, with related examples discussed and programmed in (Sapaty, 1999, 2005).

6 MORE MANAGEMENT ISSUES

In a similar way, it is easy to describe and execute much more complex spatial scenarios of both external supervision and self-organization of affected regions. These may include massive evacuation, forming new, emergency infrastructures, fighting malicious post-disaster inclusions and groupings (linked with terrorism and looting, for example), movement and spreading of external relief teams cooperating with each other and with the self-organizing disaster area, and so on. The following are some details on other important EM-related problems, which are currently being investigated with the use of WAVE-WP.

6.1 Distributed Data Mining

The currently performed data mining for a search of suspected individuals is often restricted, mostly in localized databases, as can be understood from (Markoff, 2006). But very different and mutually complementary information about, say, a suspected criminal can simultaneously be present in many databases throughout the globe, and she may have many (stable, as well as casual) links with other persons – also scattered in different databases.

The persons of interest and their groups may be on a constant move, leaving dynamically (temporary) records in different databases (say, in airports by registering their tickets, also by video cameras installed at the departures and arrivals). Links between these persons -- from direct, voice, to electronic via Internet or mobile phones -- may vary rapidly during runtime. The WAVE-WP technology provides for an effective finding and tracing of single and multiple persons/objects, discovery of any imaginable (and so far even unimaginable) patterns of possible relations between them (which may be fuzzy, dynamic, and situation dependent) throughout the whole world in a highly automated, including fully automatic mode, often ahead of time. The model does not need any central databases for this, using in parallel many computers with databases directly where they are located, without copying the information and its subsequent collection for central processing (such a collection can make dynamic information useless, actually dead).

The technology just treats the whole world with all its available databases as a parallel supercomputer/global knowledge base, and spatial programs for it can be injected from any point, covering the distributed virtual world (or its parts needed) at runtime and in parallel. No other existing approaches can compete with this so far.

6.2 Tracing Uninhabited Containers

The cargo containers traveling throughout the world may pose a serious threat due to the potential possibility of spreading unregistered hazardous objects and materials (Our porous port protections, 2006). Their great numbers, movement on large distances by different kinds of transportation means, and between and through different countries, make their supervision a serious problem for any existing systems and technologies.

With the technology described in this paper we can dynamically (and if needed, secretly) attach an individual mobile intelligence to each container and trace its movement worldwide, in parallel with millions of other such intelligences. This can also guarantee their regular remote reporting on the cargo states and direct interaction with other such intelligences, if needed. These multiple tracking intelligences can effectively cooperate with other important EM systems, for example, those providing worldwide search for suspected individuals and organizations, as in the previous section.

All this can be organized in WAVE-WP fully automatically, without human involvement, by “infecting” the world with a powerful supervisory “supervirus”, massively penetrating into any infrastructures and services, while preserving its own global integrity and goal orientation.

6.3 Memetics

Memetics, which proclaims stable self-reproducing and propagating information structures of higher levels, or memes, analogous to genes in biology, has so far been considered by many as a doubtful and controversial discipline (Finkelstein, 2006). But it can contain, in principle, high philosophical potentials for explaining and solving complex problems in human societies, which may help, for example, understand and fight terrorism globally -- if this materializes into practical methodologies, technologies, and tools. We can assume that memes, from the engineering point of view, are distributed, evolving and dynamic spatial structures, or patterns, not observable from local points as a whole, and therefore so hardly comprehendible by single human brains. Using WAVE-WP approach, we can automatically discover, study, and comprehend spatial memes spatially, much superior to a localized human brain. Moreover, the technology can apply spatial memes spatially too, in parallel, effectively impacting societies as a whole.

We can imagine a large distributed system, say, a country or an army. A meme for us may be a persistent complex pattern with hidden self-replicating and self-evolving algorithm deeply integrated with its structure (i.e. forming an active spatial pattern). In WAVE-WP, we can describe any such pattern in a highly dense, recursive form. We can then apply this pattern to the same or any other system, starting form any point, in a parallel matching & covering mode. We can modify the whole system to fit this pattern, or change the existing patterns in it to fit the one we have just injected. We can also implant into any system some higher-level patterns (or "meta-memes") that can discover the existing spatial patterns (memes) in it, copy them spatially, wrap in the WAVE-WP recursive syntax, transfer to other parts of the system or to other systems, and unwrap there, with runtime covering and updating these systems or their parts as needed. Many related programs have been written in WAVE, efficiently working in a distributed mode via the Internet (Sapaty, 2006).

The memes, whether they really exist or not as physical or virtual entities, may in any case inspire us for a search of higher-level mechanisms and abstractions in human societies in order to influence them properly, with application in emergency management and terrorism fight too, and especially.

7 CONCLUSIONS

A dynamic and ubiquitous approach for emergency management has been proposed, based on the distributed processing and control technology WAVE-WP. Operating in a different space-time continuum than the existing systems or forces that can harm them, it can set up any needed control over the distributed worlds. The compactness and simplicity of relief scenarios in the WAVE-WP language can allow us to program them on the fly, reacting timely on rapidly changing situations. The technology can effectively convert scattered human and technical resources into operable distributed systems, autonomously solving complex self-survivability and self-recovery problems. The implementation of the technology can be easily done on any existing software or hardware platform, and the corresponding "social" module, as a software or hardware wave language interpreter, can be readily installed in many wearable devices, and first of all mobile phones and handheld and laptop PCs.

As WAVE-WP technology allows us to set up the formal description of what was considered predominantly human activity before, the relief scenarios can be effectively performed by any combination of multi-agent (Sapaty, Filipe, 2005) manned and unmanned units. With this approach, humans and robots will not only perform complex relief operations together, but will also share responsibility for key decisions in different space-time locations. Massive use of advanced mobile robotics in the future relief missions, and its integration with manned components under the unified command and control, can be drastically simplified by this flexible control model too.

In this paper, only the spatial WAVE-WP automaton and examples of programming in it with relation to emergency management problems have been briefly described. Of course, the EM managers will be relieved from most of this internal relief mission programming "kitchen", as there will be user-friendly interactive facilities with graphics, video, and audio interfaces. But the technology users should also be taught how to think and act like generals rather than privates on the disaster "battlefields", in order to reduce the necessity of keeping special qualified system programmers, and quickly react in the emergency situations themselves.

REFERENCES

3G industry optimistic for 2006, 2005. *BBC NEWS, Technology*, 25 November, http://news.bbc.co.uk/.

Bullock, J. A., Haddow, G. D., 2004. The future of emergency management. *Journal of Emergency Management*, Volume 2, Number 1, pages 19-24. http://www.pnpco.com/.

Finkelstein, R., (Compiler & Editor), 2006. A Memetics Compendium: Vol. 1, 2. In *Workshop on Memetics*. Executive Conference Center of Strategic Analysis Inc., Arlington, Virginia.

GPS mobile phone, 2006. http://www.easydevices.co.uk/pcp/Mobile_Phone.html.

Joint robotics program, master plan FY 2005, 2005. US Department of Defense. http://www.jointrobotics.com/activities_new/masterplan.shtml.

LaFraniere, S., 2005. Cellphones catapult rural Africa to 21st century. *The New York Times*, August 25. www.NYTimes.com.

Markoff, J., 2006. Taking spying to higher level, agencies look for more ways to mine data, *The New yor Times*, February 25. www.nytimes.com.

Mobiles head for sales milestone, 2005. *BBC NEWS, Technology*, 19 July. http://news.bbc.co.uk/.

National Infrastructure Protection Plan, Draft, 2005. US Department of Homeland Security, Office of the Assistant Secretary for Infrastructure Protection, 161p. www.dhs.gov.

National Response Plan, 2004. US Department of Homeland Security, 95p. http://www.dhs.gov/interweb/assetlibrary/NRPbaseplan.pdf.

Nunn, S., 2005. Preventing the next terrorist attack: the theory and practice of homeland security information systems. *Journal of Homeland Security and Emergency Management*, ISSN 1547-7355, Vol. 2, Iss. 3. http://www.bepress.com/jhsem/.

Our porous port protections, 2006. *The New York Times*, Editorial, March 10. www.nytimes.com.

Samsung i730, 2005. http://www.vzwshop.com/samsungi730info/.

Sapaty, P. S., 1999. *Mobile Processing in Distributed and Open Environments*, John Wiley & Sons, ISBN: 0471195723. New York, 436 p.

Sapaty, P., Sugisaka, M., Mirenkov, N., Osano, M., Finkelstein, R., 2005. Grasping the distributed entirety. In *AROB 10th. The Tenth International Symposium on Artificial Life and Robotics*. Beppu, Japan.

Sapaty, P. S., 2005. *Ruling Distributed Dynamic Worlds*, John Wiley & Sons, ISBN 0-471-65575-9. New York, 256p.

Sapaty, P., Filipe, J., (Eds.), 2005. *Multi-agent robotic systems. Proceedings of the 1st International Workshop on Multi-Agent Robotic Systems MARS 2005*. Barcelona, Spain, September 2005, INSTICC PRESS, ISBN: 972-8865-34-1, 200p.

Sapaty, P., Sugisaka, M., Finkelstein, R., Delgado-Frias, J., Mirenkov, N., 2006. Emergent societies: an advanced IT support of crisis relief missions. In *AROB 11th. The Eleventh International Symposium on Artificial Life and Robotics*, Beppu, Japan.

Sub-$100 Laptop Design Unveiled, *BC NEWS, Technology*, 29 September 2005, http://news.bbc.co.uk/.

Wave system public domain, 2003. http://www.ece.ubc.ca/~sergiog/wavefiles/.