Interaction Design for Web Emergency Management Information Systems

Teresa Onorati
Departamento de Informática
Universidad Carlos III de Madrid, Spain
tonorati@inf.uc3m.es

Alessio Malizia
Departamento de Informática
Universidad Carlos III de Madrid, Spain
amalizia@inf.uc3m.es

Paloma Díaz
Departamento de Informática
Universidad Carlos III de Madrid, Spain
pdp@inf.uc3m.es

Ignacio Aedo
Departamento de Informática
Universidad Carlos III de Madrid, Spain
aedo@inf.uc3m.es

ABSTRACT

The interaction design for web emergency management information systems (WEMIS) is an important aspect to keep in mind due to the criticality of the domain: decision making, updating available resources, defining a task list, trusting in proposed information. A common interaction design strategy for WEMIS seems to be needed but currently there are few references in literature. Our aim is to contribute to this lack with a set of interactive principles for WEMIS. From the emergency point of view, existing WEMIS have been analyzed to extract common features and to design interaction principles for emergency. Furthermore, we studied design principles extracted from the Turoff's model relating them to emergency phases and features. In particular, in this paper, we choose to follow the current trend in the definition of emergency life cycles. In our approach, referring to general policies in literature, the emergency management process is divided into two different sub cycles: back-end and front-end. From the interaction point of view, a formalization process based on the interactive PIE model has been defined. The result we propose here is a set of design principles for supporting interactive properties for WEMIS.

Keywords

Design Methodologies for Human-Computer Interaction, Emergency Life Cycle for Decision Support, Models for Communication and Information Interaction

INTRODUCTION

The management of information and resources is a crucial activity for the agencies and organizations managing emergency situations. The scope of Emergency Management Information Systems (EMIS) is to support the activities usually performed by emergency workers: they are focused on the organization of available information and resources (Van de Walle and Turoff, 2007). Emergency management is a collaborative and multi-organizational endeavor [Waugh and Streib, 2006] and, therefore, EMIS have to provide communication channels and collaborative tools for teams that are not only geographically distributed but also functionally independent. By functionally independent we mean that different teams might have different goals, perspectives and operation protocols. Nowadays, many agencies and organizations for emergency management use web-based EMIS (WEMIS) as a support tool for the cooperation and the coordination in different emergency phases. WEMIS employ Internet protocols and facilities for the communication during coordination activities. Eventually, they do not require an installation procedure and they have a high portability: all devices with an Internet connection can access to them. In this way, also people that work in direct contact with the emergency area, as firemen or policemen, can use them with mobile devices and communicate with coordination offices.

In this paper, we focus on WEMIS that make it possible to efficiently communicate and share information. More specifically, our interest lies on the design of the interaction with WEMIS. Emergency workers have to manage complex situations where short time decision making is fundamental: victims and damages depend on the emergency solution strategy. For this reason, users need a quick and trustworthy interaction with the system: they have to know exactly the next task to perform, which kind of results and consequences to expect, and which information and status have to be updated. In other words, the WEMIS has to support at some level

Reviewing Statement: This paper has been fully double blind peer reviewed.

situation awareness [Endsley, 2000] by providing accurate, timely and appropriate information at each stage so that each user can understand the situation (situation assessment) and decide how to react properly. Considering emergency management as a team work, we should transcend the concept of situation awareness to move onto activity awareness which is the "*awareness of project work that supports group performance on complex tasks*" [Carroll et al, 2003]. Even though no system can guarantee situational or activity awareness, a good interaction design can help to support it by ensuring that users will have the *right information at the right time and in the right form* both concerning the situation as well as the performance of other cooperating teams.

The design of the interaction with the WEMIS is therefore a cornerstone as systems should support users in developing their tasks without interfering with their usual protocols or imposing any kind of burden (Carver and Turoff, 2007). Moreover, the adoption of this kind of systems is not straightforward as many factors influence their real usage in emergency situations (Aedo, Diaz, Carroll, Convertino and Rosson, 2009).For instance, the subjective impressions of users about their personal capability to use the system, the degree of control, the support from other colleagues or from the organization, have a strong influence in the final decision to use a system (Mathieson, Peacok and Chin, 2001). The interaction design process for WEMIS needs to address all these issues properly: a possible approach could be the participatory design where users and stakeholders are actively involved.

There are some design principles like those defined in (Turoff, Chumer, Van de Walle and Yao, 2004) but until now, interactive design models have not been specifically defined for the emergency domain and in particular for WEMIS. The aim of our work is to contribute to this lack in the definition of a set of interactive principles specific for WEMIS. In the first section, we present a survey about existing WEMIS to find out common characteristics and design aspects. In the second section, we categorized WEMIS features into twenty-three different classes relating them with emergency supported by the framework proposed by Chen, Sharman, Rao and Upadhyaya (2008). In the third section we present our main contribution, a set of interactive properties for WEMIS. We relate features' classes, as previously defined, to a set of design principles extracted from a specific design model for the emergency management domain: DERMIS (Turoff, Chumer, Van de Walle and Yao, 2004). Finally, these principles have been formalized using the PIE model (Dix, Finlay, Abowd and Beale, 2004), obtaining a number of formal interactive properties fulfilled if our design principles are employed.

A STUDY OF EMERGENCY MANAGEMENT INFORMATION SYSTEMS IN THE WEB

The main basis for the design process of existing WEMIS is the analysis of data collected from real experiences in emergency management. For instance, the Sahana web system has been developed from the observation of the emergency management process applied during the tsunami of 2004 in Sri Lanka (De Silva, De Silva, Careem, Raschid and Weerawarana, 2006); similarly SIGAME was based on the lessons learned from the massive wild fires in Galicia in 2007 (Aedo, Díaz and Díez, 2009). However, apart from individual experiences there is a lack of clear definition of the list of interactive tasks to perform for emergency planning and solving. Moreover, until now a common and formal design strategy has not been defined yet to develop WEMIS. In this section we analyze a set of WEMIS to extract common and frequent characteristics.

The set of WEMIS that we considered has been extracted from the W3C Working Group[1], called Emergency Information Interoperability Framework (EIIF), which aim is to define standards for emergency management domain. In Table 1, there are the eight WEMIS that we have considered from the W3C classification. For each of them we have collected a detailed description about the management process, available features, used standards and real use cases. All these data have been extracted both from the W3C classification and by studying system's references. Table 1 summarizes the main features of these systems.

WEMIS	Website	Main Features
Sahana Disaster Management System	www.sahana.lk	Free and Open Source Software // Tracing // Inventory Management // Aid Distribution // Situation Mapping // Shelter tracking // Responder Management
UN OCHA Who is Doing What Where	3w.unocha.org	Mapping // Flexible architecture // Synchronization
UN Reliefweb	www.reliefweb.int	Mapping // Funding // On-line library // Professional Resources // Web feed service
LA RED/ UNDP-GRIP DesInventar	www.desinventar.net	Disaster datacard // Database Support// GIS systems // Google maps and Google Earth tools // Excel and XML

[1] Emergency/Disaster Management Systems and Products, www.w3.org/2005/Incubator/eiif/wiki/EMSystems, last access October 2009.

		formats
Federal Emergency Management Information System	www.pnl.gov/FEMIS	Resource tracking // Task lists // Contact lists // Event logs // Status boards // Hazard modeling // Evacuation modeling
UN GDACS Virtual OSOCC	www.gdacs.org	Disaster alerts // Situation overview // List and status of emergency teams // Satellite based maps
(FEMA) National Shelter System	www.fema.gov	Traditional shelters // Household Pet // Medical Special Needs // Kitchens // Points of Distribution // Food warehouses // GIS mapping
SIGAME	www.sigame.es	Coordination between communities // Coordination of the supra-communitarian aids // Monitoring of resources

Table 1. Considered WEMIS for the analysis

From our study we can conclude that there are some common features, but the design and the implementation is usually completely different. For example, in Sahana the mapping of available resources and organizations is a textual list with the identification of the data and the location whilst in DesInventar it is represented as an interactive map. Another example is the notification of disaster information: in GDACS it is shown directly on the homepage, instead in FEMA it is in a different page accessible from the main menu.

In order to obtain a common list of features for WEMIS that could help to define a common interaction model, in the next section we are going to group activities in Table 1 into general classes of functionalities.

A CATEGORIZATION OF EMERGENCY CHARACTERISTICS BY PHASES

The emergency management process involves a complex set of activities, each of which has a specific scope and requires different kinds of resources, information, cooperation degree and worker expertise. Moreover, an important dimension to consider is timing: the emergency process can be seen as a sequence of stages where the tasks to be performed strongly depend on specific requirements of the emergency phase.

In literature, emergency phases have been identified in different ways. Waugh and Streib (2006) identified four phases: (1) prevention and mitigation; (2) planning and training; (3) response; (4) recuperation. Another framework based on a lifecycle approach is the emergency response coordination lifecycle presented in (Chen et al., 2008). Compared to (Waugh et al., 2006), this framework introduces a distinction between activities related to front-end situations and those concerning back-end organizations. In this way it is possible to specify a task list depending on the role and responsibility of workers. For example, front-end workers use different kinds of devices compared to back-end workers: applying the Chen's framework it is possible to face this difference with an appropriate design strategy. For instance, in (Catarci, T., Leoni, M.D., Rosa, F.D., Mecella, M., Poggi, A., Dustdar, S., Juszczyk, L., Truong, H.L. and Vetere, G., 2007), authors presents the European project Workpad: a 2-level framework for the collaboration among emergency teams that confirms the current trend to manage emergencies distinguishing between back-end and front-end communities. The scope is to provide specific services for each community. For example, the back-end community needs an efficient way to organize information collected about the disaster, like maps, resources and localization of emergency points. The front-end community, on the other end, needs a fast communication channel to receive back-end information and organize the emergency response. In (Chen et al., 2008), the life cycle is divided into three stages: pre-incident, during-incident, post-incident. The first one concerns with activities related to planning, training and organizing involved teams. The response management of occurred crisis and the coordination of workers are part of the during-incident phase. Furthermore, the during-incident stage is viewed as the composition of two sub-cycles: mini-second and many-second. The mini-second coordination cycle is the onsite organization of teams and front-end emergency workers for a rapid intervention and prevention. The many-second coordination cycle concerns the emergency operations center (EOP) and all supervisor structures for an efficient communication with the onsite response. The post-incident or recovery phase focuses on the return to a normal situation and the analysis of collected information in order to improve future planning. Moreover, the framework defines five basic elements and it applies them to each stage of the coordination phase: (1) *task flow* with tasks to perform and relations among them; (2) *resources management*; (3) *information management*; (3) *decision making* about organization and structure; and (5) *responders* involved to complete the task flow.

We categorized the set of features presented in Table 1 to fit general classes of features, and thus, summarizing a set of common features into a general category; for example: *mapping*, *situation mapping*, *Google maps and Google Earth tools* taken from Table 1 all fit into the *Spatial Registry* feature class shown in Table 2.

By relating features classes and different emergency phases within the Chen's lifecycle framework we propose an organization of information that before was spread among different aspects of WEMIS. The result is shown in Table 2: we have defined eighteen classes structured into three main emergency phases.

For pre-incident response, we have individuated five classes for planning, training and modeling information. The first class is the *Organization Registry* for the management of organizations and teams working in the disaster area, including personnel information about each worker and its responsibilities and capabilities, so that the WEMIS can have data to decide which are the tasks that can be assigned to each worker and the kind of information he would need to improve his situation assessment. The second one is the *Spatial Registry* to establish a relation between emergency information and geographic localizations, so distributed teams as well as resources can be organized in an efficient way. The third class is the *Emergency Message Center* to define a communication channel for planning meetings and trainings. Next, the *Hazard and Vulnerability Modeling* concerns the modeling of information collected during past disasters to evaluate the risks and vulnerabilities. The last class is the *Demographic Registry* that manages people living or working in disaster areas.

For the mini-second coordination cycle during the incident response, we identified four classes and main features about tracking tasks and managing communications. The *Task Tracking* class keeps track of performed tasks, so that support for activity awareness can be provided, collecting information from front-end teams. Next, the *Victims Registry* manages affected people, updating lost, dead and injured. The third class is the *Emergency Teams Registry* to follow activities of each team. The last one is the E*mergency Message Center*, the same one we have defined for the pre-incident stage, but in this case the communication channel is established among front-end teams to share information and resources.

Activities for the many-second coordination cycle during the incident response are grouped into eight classes related to a higher level of management. The *Task Tracking* is for main operations and decision-making, where an operation is a set of simple tasks. The *Victims Registry* is defined crossing different affected areas. In the *Emergency Message Center*, the communication channel is established among back-end workers to coordinate operations and organizations. The *Collaborative Suggestions Collection* collects information and suggestions from emergency teams in order to choose the best-fitting plan and task list. Next, several registries are defined about *Requests*, *Shelter* and *Volunteer* to keep updated available information. The last class is the *Situation Mapping* to monitor the general response process, collecting information from all involved teams and performed operations.

Finally, in the post-incident response we have defined six classes related to the improvement of future emergency response. In this case the *Task Tracking* class presents final results for performed tasks: this information can be employed for next analysis, as *Improvement*, *Estimation of population at risk* and *Report Registry*. Moreover, a *Discussion Forum* supported by a *Photo Library* is created among organizations and emergency workers for a collaborative exchange of opinions and best practices with a view to elicit knowledge sharing.

Stage		**General Features**
Pre-Incident Response		Organization Registry: collect information about involved organizations in the disaster region.
		Spatial Registry: associate emergency information with geographic information system
		Emergency Message Center for training and meetings
		Hazard and vulnerability modeling
		Demographic Registry
During Incident Response	Mini-Second Coordination Cycle	Task Tracking for front-end teams
		Victims Registry
		Emergency Message Center for front-end teams
		Emergency Teams Registry for front-end teams
	Many-Second Coordination Cycle	Emergency Message Center for back-end workers to coordinate all involved organizations
		Task Tracking for operations and decisions
		Collaborative Suggestions Collection about the task list
		Requests Registry
		Shelter and Resource Registry
		Victims Registry
		Volunteer Registry

		Situation Mapping
Post-Incident Response		Task tracking with final results and real processes
		Discussion forum among involved organizations and emergency workers
		Photo library
		Analysis and Improvement for task tracking
		Reports Registry
		Estimation of population at risk for future pre-incident planning

Table 2. Emergency coordination life cycle and classes of features

In the next section, we derive from this categorization a set of design principles and formalized interactive properties.

FROM DERMIS TO PIE MODEL: INTERACTIVE DESIGN PRINCIPLES FOR WEMIS

This section presents our proposal to structure the interaction in the WEMIS domain: from the emergency life cycle to formal properties for interaction design principles. The definition of our interaction design principles has two different stages. During the first one, the relation between emergency phases, features and design principles is established to derive design principles from existing practices. In this case, we are going to merge the results obtained in Table 2 and the Dynamic Emergency Response Management Information System (DERMIS) model presented in (Turoff et al., 2004). During the second stage, emergency design principles are formalized through the PIE (Program, Interpretation, Effect) formalism (Dix et al., 2004) to obtain a set of interactive properties related to phases and features. We will briefly describe here the DERMIS and PIE model; for an extended description see Appendix A.

In (Turoff et al., 2004), authors present a framework for the design of EMIS, called DERMIS model. From the analysis of organization and planning of real emergency management agencies, the DERMIS model has been structured into four different sections: *Design Premises*, *Conceptual Design*, *General Design Principles and Specifications*, and *Supporting Design Considerations and Specifications*. We have formalized and specialized for interaction the set of general design principles by using the PIE model. The scope of the PIE model is to structure the interaction between users and systems (Dix et al., 2004), through a formal language. The interaction starts with a set of user's actions as input: each action is called *command* (C) and a sequence of commands is represented as a *program* (P). The output of the system is the result obtained from the execution of the program and it is called *effect* (E). The sequence of system functions for generating the output is represented by an interpretation function I that relates programs with effects. For example, if a user wants to print a file, the print function is the program and the printed document is the effect. Furthermore, the effect generates a set of *results* (R) and a set of *observable effects* (O). Results represent the system output (e.g. the printing of the document); instead, the observable effect is what the screen visualizes (e.g. the editor and the open document). At last, if the screen cannot visualize the entry observable effect, but just a part of it, this part is the *display* (D) (e.g. the display is the part of a document currently visualized in the screen and the observable effect is the entire document that can be visualized with the scrolling). Programs, effects, results, observable effects and displays are the main elements of the PIE.

THE ENRICHED EMERGENCY LIFE CYCLE: INTERACTION DESIGN PRINCIPLES FOR WEMIS

We have enriched the emergency life cycle with interactive principles defined through two different phases. During the first one, we have related principles from the DERMIS model to emergency features and phases. In particular, for each stage of the coordination life cycle, we have considered individuated classes of features (see Table 2) in order to recognize appropriate DERMIS principles. For example, let us consider the during-incident response stage and the mini-second coordination cycle (see Table 3). The *Task Tracking* and the *Victims Registry* features can be related with the *Information Source and Timeliness* and the *Up-to-date Information and Data* principles. All of them are about the visualization and the updating of information: in order to perform an efficient task and victims tracking it is important to provide appropriate data in a well-structured way to facilitate the access. The second class of features is the *Emergency Message Center for front-end teams*. In this case, related principle is the *Open Multi-Directional Communication* to structure the communication of updated information. The last class is the *Emergency Teams Registry*. Here, we need the *Authority, Responsibility, and Accountability* to support the definition of different levels of responsibility and the social interaction among workers.

During the second phase for the definition of the enriched life cycle, we have formalized DERMIS principles through the PIE model properties. In particular, each principle has been formalized in order to obtain the proper interactive property among four possibilities: *observability*, *reachability*, *predictability*, *transparency* and *meta-communication*. While observability, reachability, predictability and transparency are already defined within the PIE model (see Appendix A), our contribution also consists of introducing *meta-communication* (de Souza, 2004) to represent rules and semantics for the communication between users and designers, so that participatory design can be supported. A high meta-communication corresponds to a clear understanding of the designer's strategy by users: in this way, users can access to available information through an efficient interaction. The reachability is implicitly defined into the observability and predictability as the possibility to reach a specific state of the system from any other through a sequence of actions. Moreover, observability and predictability represent the basis to define transparency and meta-communication. The observability is referred to visualized information on the screen: a system is observable if users can understand completely the output of the system observing the display. The predictability is similar to the observability, but for future states: a system is predictable if observing the display users are able to determine which states will be reached by the system through future actions. Users of an observable and predictable system know exactly the current and future results they will obtain from running functionalities. In Appendix A, there is a detailed description of each interactive property. From these definitions, now we can derive the transparency as a combination of predictability and observability: a transparent system visualizes clearly on the screen all information and data about performed operations, events and effects. In this way, users can know exactly current and future states, avoiding possible mistakes. As confirmed by the formalization process, in the emergency management domain, the transparency represents the solution to ensure that users get updated and relevant information and an efficient exchange of messages, so that it might contribute to develop an appropriate situational awareness.

In Figure 1, there is a graphical representation of the extended PIE model. There are two actors: the designer and the user. Each one defines an interpretation function (respectively, I_D and I_U) between sequences of actions (respectively, P_D and P_U) and expected system states (the effect E). Other involved elements are the available information (the observable effect O), what is now visible in the screen (the display D) and the system's output (the result R). Furthermore, the defined interactive properties are represented as functions among these elements: *observe* between E and O for observability, *result* between E and R for predictability, *transparent* among E, O and R for transparency and *m-communication* between designer and user for meta-communication. In Appendix A there is a formal definition of the elements and properties introduced here.

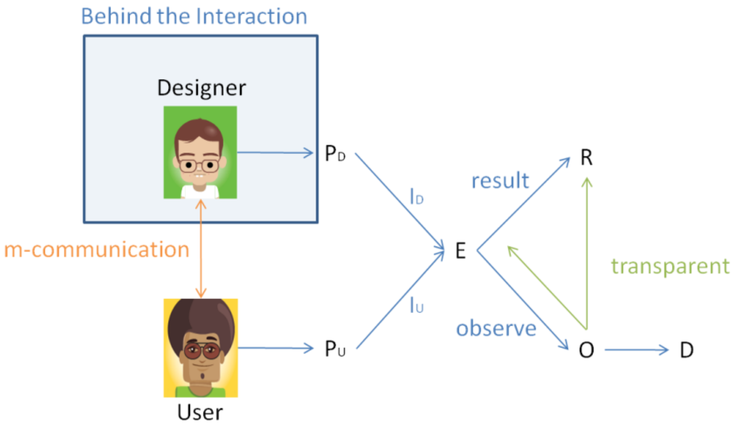

Figure 1. Extended PIE model with interaction properties

In Table 3 there is the life cycle we have obtained at the end of the formalization for the mini-second coordination cycle in during-incident response stage. The first principle we have analyzed is the *Information Source and Timeliness*. Representing the information as a program (P) and its current state as an effect (E), the principle can be traduced displaying all available data of the state (the result R of the effect E): this corresponds to the transparency property.

The second principle is the *Up-to-Date Information and Data*. In this case we can recognize the *gone away for a cup of tea* metaphor: the requirement is that users could understand and easily access information through the interface, even if they are distracted by external factors. Employing the same annotation of the previous principle, this means that observing the state of the information (E) from the screen (O), all available data about the information (R) are clearly visualized: this corresponds to the predictability.

Stage		General Features	DERMIS Design Model (Turoff) Design Principles	Interactive Principles
During Incident Response	Mini-Second Coordination Cycle	Task Tracking for front-end teams & Victims Registry	***Information Source and Timeliness*** – *In an emergency it is critical that every bit of quantitative or qualitative data brought into the system dealing with the ongoing emergency be identified by its human or database source, by its time of occurrence, and by its status. Also, where appropriate, by its location and by links to whatever it is referring to that already exists within the system*	***Information Source and Timeliness*** → a complete representation of information → considering information as a program P and its visualization as an effect E, we visualize the complete status of available information → transparency $\forall e \in E \,.\, predict(observe(e)) = (e)$, where $\forall e \in E \,.\, predict(observe(e)) = result(e)$
			Up-to-date Information and Data – *Data that reaches a user and/or his/her interface device must be updated whenever it is viewed on the screen or presented verbally to the user*	***Up-to-date Information and Data*** → "gone away for a cup of tea" metaphor → predictability $\forall e \in E \,.\, predict(observe(e)) = result(e)$
		Emergency Message Center for front-end teams	***Open Multi - Directional Communication*** – *A system such as this must be viewed as an open and flat communication process among all those involved in reacting to the disaster*	***Open Multi - Directional Communication*** → all users have to be aware of the current state of the system, from information to communication → considering the information as a program (P) and its visualization as an effect (E), we want to communicate P in order to avoid misunderstanding about E → transparency and meta-communication $\forall e \in E \,.\, predict(observe(e)) = (e)$ & $\forall e \in E \,.\, I_D^{-1}(e) = I_U^{-1}(e)$
		Emergency Teams Registry for front-end teams	***Authority, Responsibility, and Accountability*** – *Authority in an emergency flows down to where the actions are taking place*	***Authority, Responsibility, and Accountability*** → we have to adapt the system to different roles and users that interact with it → improve the meta-communication between designer and user → meta-communication $\forall e \in E \,.\, I_D^{-1}(e) = I_U^{-1}(e)$

Table 3. The enriched emergency life cycle for mini-second coordination

The third principle we are going to present here is the *Open Multi-Directional Communication* about the updating of information and the communication process to exchange this information among people. In this case, we have formalized two different properties: the transparency and the meta-communication. The transparency is obtained in the same way we have shown for the *Information Source and Timeliness* principle. For the communication channel, users have to receive messages without misunderstanding problem. Considering the message to send as a program (P_D), the received message by the users as another program (P_U), the scope of the communication as an effect (E), the principle can be formalized requiring that given E, P_D and P_U have to be the same, as in the meta-communication property.

Finally, the last principle for the mini-second coordination cycle is the *Communication Authority, Responsibility, and Accountability*. The idea is to adapt the system to different roles and responsibilities users might have. For this scope, it is important to understand available information and data in the screen and eventually adapt them to users' needs. Interpreting the principle in this way, we can formalize it with the meta-communication process as in the previous case. In this case, the designer has to communicate efficiently which programs and effects are accessible, depending on roles and responsibilities.

The Table 3 shows a view of the enriched emergency life cycle obtained as the result of the formalization process defined here. There are three columns: stages, general features and design principles with PIE formalism. The stages column contains emergency phases as they have been defined in (Chen et al., 2008). The general features column has the classes we have defined in Table 2. The last column concerns the design principles that we have extracted from the DERMIS model and the formalization process applied to define interactive properties. While, the DERMIS model identifies important characteristics for emergency management systems and gives to designers guidelines to implement them, our properties are related to interactive aspects of the relation between users and systems, designers and systems and users and designers. As we have shown in Table 3, we have extracted a set of interactive properties with a bottom-up process, where design principles from DERMIS model represent the first step for the formalization of emergency features. The PIE model, instead, has been used as formalism for the final step to obtain the properties. Moreover, we have applied an extended PIE model, where we have added the meta-communication as interactive property between users and designers.

CONCLUSION AND FUTURE WORKS

In this paper, we have presented a first contribution in the definition of interaction design principles for WEMIS. From the analysis of existing WEMIS, we have found out that there is no unique definition of interaction design for the emergency management process. For this reason, as a first contribution in Table 2 we have defined the relation between emergency phases and WEMIS features' classes. Successively, we have related them to a set of design principles extracted from the DERMIS model. Finally, through collected information about emergency phases and WEMIS features, we have formalized these principles using the PIE model and we have obtained a set of interaction design principles for WEMIS guaranteeing such systems properties: observability, predictability, reachability, transparency and meta-communication.

The next step will be the definition of formal patterns for the interaction design of WEMIS. The idea is to solve design critical aspects collected during the WEMIS and literature analysis with the application of interactive properties presented in this paper. Designers might use these patterns for designing web based systems in the emergency domain in order to guarantee an efficient interaction (by supporting the properties obtained by the application of such design principles). Finally, we plan to test these interaction patterns in real case studies.

ACKNOWLEDGMENTS

This work is funded by projects UIA4SIGE (Ministry of Science and Innovation TSI2007-60388)

REFERENCES

1. Aedo, I., Díaz, P., Carroll, J. M., Convertino, G., and Rosson, M. B. (2010) End-user oriented strategies to facilitate multi-organizational adoption of emergency management information systems. *Inf. Process. Manage*. 46, 1, 11-21.
2. Aedo, I., Díaz, P. and Díez, D. (2009) Cooperation amongst autonomous governmental agencies using the SIGAME web information system. *Proceedings of EGOV 2009*.
3. Carroll, J.M., Neale, D.C., Isenhour, P.L, Rosson, M.B. and McCrickard, D.S. (2003). Notification and awareness: synchronizing task-oriented collaborative activity. *International Journal of Human-Computer Studies* 58, 605-632.
4. Catarci, T., Leoni, M.D., Rosa, F.D., Mecella, M., Poggi, A., Dustdar, S., Juszczyk, L., Truong, H.L. and Vetere, G. (2007) The WORKPAD P2P Service-Oriented Infrastructure for Emergency Management Enabling Technologies: Infrastructure for Collaborative Enterprises, *Proceedings of the16th IEEE International Workshops WETICE*, 147 - 152
5. Carver, L, and Turoff, M. (2007). Human Computer Interaction: The human and computer as a team in emergency management information systems. *Communication of the ACM*, 50, 3, 33-38.

6. Chen,R., Sharman, R., Rao, H. R. and Upadhyaya, S. J. (2008) Coordination in Emergency Response Management, *Communications of the ACM*, 51, 5, 66-73.
7. De Silva, C., De Silva, R.,Careem, M., Raschid, L. and Weerawarana, S. (2006) Sahana: Overview of a Disaster Management System, *Proceedings of the IEEE International Conference on Information and Automation (ICIA)*.
8. de Souza, C., (2004) The Semiotic Engineering of Human-Computer Interaction, Ma: the MIT Press, Cambridge.
9. Dix, A.J., Finlay, J.E., Abowd, G.D. and Beale, R. (2004) Human-Computer Interaction (3rd edn), Prentice-Hall, London.
10. Endsley, M. R. (2000). Theoretical underpinnings of situation awareness: A critical review. In M. R. Endsley & D. J. Garland (Eds.), Situation awareness analysis and measurement. Mahwah, NJ: LEA.
11. Turoff, M., Chumer, M., Van de Walle, B. and Yao, X. (2004) The Design of a Dynamic Emergency Response Management Information System (Dermis),*Journal of Information Technology Theory and Application (JITTA)*, 5, 4, 1-36.
12. Van de Walle, B. and Turoff, M. (2007) Emergency Response Information Systems: emerging trends and technologies, *Communications of the ACM*, 50, 3.

APPENDIX A

The DERMIS Design Model

In (Turoff et al., 2004), authors present a framework for the design of DERMIS systems, called DERMIS model. It has been developed collecting data from real emergency management experiences in USA. In particular, authors have considered the Office of Emergency Preparedness (OEP) and the Federal Emergency Management Agency (FEMA).

The first section is a collection of nine premises that designers have to accomplish in an initial phase of the design process: for example, the first one is titled *System Training and Simulation* and underlines the importance of training during the pre-incident phase.

The second section, *Conceptual Design*, presents five criteria to design the interface: *Metaphors*, *Human Roles*, *Notifications*, *Context Visibility* and *Hypertext*. They adapt human-computer interaction concepts to the emergency domain. For example, *Human Roles* represent groups of users that perform same activities: roles are defined depending on the level of responsibility, required tasks and emergency phases.

The third section is the *General Design Principles and Specifications*. There are eight principles: *System Directory*; *Information Source and Timeliness*; *Open Multi-Directional Communication*; *Content as Address*; *Up-to-date Information and Data*; *Link Relevant Information and Data*; *Authority, Responsibility, and Accountability*; *Psychological and Social Needs*. These principles are general guidelines for designers: they are more specific than other sections of the model and for this reason they will be applied during the first step of our definition of interactive properties. The first principle is *System Directory* that guides the designer in structuring data and information with a hierarchical tree for a useful text search. Other principles related to the management of available information are the *Information Source and Timeliness*, the *Content as Address*, the *Up-to-date Information and Data* and the *Link Relevant Information and Data* that are about the usage of database, the definition of semantic links and addresses, and the visualization of last updating in the interface. Moreover, the *Open Multi-Directional Communication* defines the entire emergency response process as an open communication among involved people and the *Authority, Responsibility, and Accountability* assigns responsibilities. The last guideline is the *Psychological and Social Needs* of emergency workers and affected people during a disaster.

The last section is *Supporting Design Considerations and Specifications* with general requirements and support tools: databases, collective memory to manage relations and dependencies among events, and online communities of domain experts.

The PIE Model

The PIE model is an interaction model presented in (Dix et al., 2004): it presents a set of interactive properties through a formal language based on the fist-order-logic. This model defines five elements: a *program* (P), an *effect* (E), a *result* (R), an *observable effect* (O) and a *display* (D). The interaction among these elements is represented through five properties: observability, predictability, reachability, transparency and meta-communication. While observability, predictability, reachability and transparency are already defined in the PIE model, the meta-communication is an additional property that we have introduced in the model. Next, we are going to present a formal description of these interactive properties.

The observability represents what is displayed on the screen about the effect: this means that here we are measuring how many data users can know about the effect just observing the screen. In particular, a system is observable if users can understand completely the output of the system (effect) from information visualized on the screen (observable effect). The formalization of the property is:

$$\exists \text{ observe}: O \rightarrow R \text{ . } \forall \text{ e} \in E \text{ . observe(e)} = \text{e} \quad ,$$

where observe is a function from the set of observable effects O to the set of results R and it is defined for each effect in E.

The predictability is related to the observability, but for future states: observing the screen, users must be able to determine future states of the system. Another way to explain this property is through the metaphor of *gone away for a cup of tea problem*, also presented in (Dix et al., 2004). To illustrate this metaphor, consider a user that is working on a program and decides to leave her tasks to get a cup of tea. When she comes back, in order to continue efficiently her tasks she needs to remember the command history and which action will be the next to perform. A predictable system displays useful information on the screen about past and future commands to help the user in resuming her work. Formally, the predictability is expressed as:

$$\exists \text{ predict}: O \rightarrow R \text{ . } \forall \text{ e} \in E \text{ . predict(observe(e))} = \text{result(e)} \quad ,$$

where predict is a function from the set of observable effects O to the set of results R and it is defined for each effect in E through the observability property (observe(e)).

The reachability represents the possibility to reach any states of the system from one of them. In this way, users should have undo facilities in order to come back and eventually correct mistakes. The formal definition of this property uses the *doit* function for the transaction from a state to another one through a sequence of commands. The formalization is:

$$\forall \text{ e, e'} \in E \text{ . } \exists \text{ p} \in P \text{ . doit(e, p)} = \text{e'} \quad ,$$

where e and e' are effects in E, P is the set of programs and the transaction function doit is defined from couple (effect, program) to a new effect.

The transparency corresponds to a stronger kind of predictability: if the system is fully predictable, it is transparent too. The screen displays all information and data about the state of the system and users can know exactly current effects. Moreover, the transparency allows predicting future state and avoiding possible mistakes. The formal form is:

$$\exists \text{ transparent}: O \rightarrow E \text{ . } \forall \text{ e} \in E \text{ . predict(observe(e))} = \text{(e)} \quad ,$$

where transparent is a function from the set of observable effects O to the set of effects E, through the observability and the predictability (predict(observe(e)) properties.

The meta-communication represents rules and semantics for the communication between users and designers. In order to introduce this additional property, we have extended the PIE model with two new actors, the designer and the user. Next, we have substituted the set of programs P with two new elements: the program defined by the designer P_D and the program defined by the user P_U. Both of them are related to the effect E through two different interpretation functions, I_D and I_U. The meta-communication is defined between these two elements. Given an effect E, P_D and P_U are the user and designer's programs to reach it. If they are composed by the same sequence of commands, the designer has communicated correctly her design strategy to the user and the meta-communication is high. Otherwise, if P_D differs from P_U, the communication between designer and user does not work properly and the level of meta-communication is low. The formal form is:

$$\exists \text{ m-communication}: E \rightarrow P \ \forall \text{ e} \in E \text{ . } I_D^{-1}(\text{e}) = I_U^{-1}(\text{e}) \quad ,$$

where m-communication is a function defined from the set of effects E to the set of programs P, through the inverse function of the designer interpretation I_D^{-1} and the inverse function of the user interpretation I_U^{-1}.