FireGrid: An e-Infrastructure for Next-Generation Emergency Response Support

Liangxiu Han[1], Stephen Potter[2], George Beckett[3], Gavin Pringle[3], Sung-Han Koo, Rochan Upadhyay[4], Gerhard Wickler[2], Dave Berry[1], Stephen Welch[4], Asif Usmani[4], Jose Torero[4] and Austin Tate[2]

[1]National e-Science Centre, [2]Artificial Intelligence Applications Institute, [3]Edinburgh Parallel Computing Centre, [4]BRE Centre for Fire Safety Engineering, *all* The University of Edinburgh, Edinburgh, UK

Abstract

The FireGrid project aims to harness the potential of advanced forms of computation to support the response to large-scale emergencies (with an initial focus on the response to fires in the built environment). Computational models of physical phenomena are deployed on High Performance Computing (HPC) resources to interpret live sensor data from an emergency in real-time – or, in the case of predictive models, faster-than-real time. These interpretations are accessed over a Grid from an agent-based system, of which the human responders form an integral part. This paper describes the FireGrid architecture and discusses the results of its application to a large-scale fire experiment.

Keywords: *Emergency Response, Grid, High Performance Computing, Multi-Agent System*

1 Introduction

Humans have always suffered natural and man-made disasters and calamities, including earthquakes, tsunamis, floods, fires, hurricanes, epidemics, industrial accidents and terrorist attacks. To minimise losses to life and property from disasters, decisions have to be made by humans in a timely manner; and the availability of relevant information is critical if the right decisions are to be made. Advances in Information Technology (IT) provide alternative channels for information flow, with the potential to place additional crucial information at the disposal of decision makers during an emergency response. The vision for the FireGrid project [1] is of a generic software architecture that provides an integrated IT solution primarily for supporting the response to emergency events. In the first instance, the project has focused on emergency response to fires in the built environment.

For obvious reasons, fire-fighters will rarely be aware of the exact conditions that hold within a building during a fire incident and, consequently, they will be compelled to make intervention decisions based on the information provided by their senses, on their training and on their past experience of fires. Furthermore, since fire is a complex phenomenon, the interpretation and extrapolation of its physical manifestations is a difficult task. Advances in several technologies when taken together suggest a possible solution to the problem:

- Developments in sensor technology, and a reduction in unit cost, offer the prospect of deploying large-scale, robust and cost-effective sensor networks within buildings;
- Advances in the understanding of fire and related phenomena have resulted in sophisticated computational models which might be used to interpret sensor data;
- The availability of Grid infrastructure for distributed High Performance Computing (HPC) and data processing suggests a platform on which these models could be run in faster-than-real time, making their use in emergencies a practical proposition.

The FireGrid approach aims to improve – both in range and quality – the information available to fire-fighters. The emphasis of the project lies firmly on the *integration* of existing technologies in the areas mentioned above rather than on the development of new ones (and hence, one outcome of the project should be an improved understanding of the requirements for the task, along with some indication of the extent to which existing technologies meet these requirements). In practical terms, this would involve the loose coupling of diverse computational models, seeded and steered by real-time sensor data, and processed using HPC resources accessed across a Grid. An agent-based command-and-control layer using Artificial Intelligence (AI) approaches to knowledge representation and reasoning would give end users access to the generated information. An initial architecture along these lines has been developed, and has been implemented with specific instances of the technologies in question and applied to a number of real fires.

The FireGrid project is one among many projects and initiatives that have been devoted in recent years to IT support for emergency management. Related work includes Geographical Information System-based applications, developed for decision makers to analyze, manage and respond to emergency situations [2][3][4]. AI has been applied to, for example, a knowledge-based model approach to flood emergency decision-support [5], and a multi-agent system has been proposed as a means of providing information to emergency planners and responders (see, for example [6]). Notwithstanding the amount of previous work in this field, FireGrid is unique in terms of the technologies it considers, its approach to integrating them and in the challenges it faces. The current paper aims to provide an overview of the project, and is structured as follows. Section 2 describes the nature of simulation models, which influences much of the work presented here; section 3 describes the developed architecture; section 4 describes an implementation of this architecture and its use during a large-scale fire experiment; section 5 discusses the results of this experiment and the findings of the project more widely, and section 6 provides some concluding remarks.

2 Simulation Models and Their Use in FireGrid

Central to the FireGrid concept is the use of simulation models of physical phenomena to provide information to emergency responders. Given the initial focus of FireGrid on fire in the built environment, our interest lies in re-using existing computational models able to interpret and predict the behaviour of the fire, the movement of smoke, the reaction of the building and its occupants during the incident, and so on. However, the use of models in such as fashion is not uncontroversial, and a consideration of the available models soon reveals a number of issues [7][8], which may be summarised as:

- *The appropriateness of the modelling approach*: hitherto the prevalent methodology in fire engineering disciplines has been to build specific models of specific fires in a *post hoc* fashion. That is, having carefully designed a fire experiment, measured all the parameters thought relevant and then conducted the fire and collected sensor data, the modelling task is then to use this information to generate the model that when 'run' most closely simulates the collected data. One should not conclude that models constructed in such a fashion will readily lend themselves to use for *ad hoc* modelling of real emergencies, where the values of certain relevant parameters will be uncertain at best and often simply unknowable. (Indeed, even for *post hoc* modelling it is often the case that values of certain parameters are difficult to ascertain with the desired accuracy.)
- *Speed versus accuracy trade-off*: the models that can run most quickly – such as course-grained analytical and zone models – are usually also those with least accuracy and reliability. Those with greatest accuracy (such as detailed CFD models) are also those that require most computing time, typically needing many hours to model an event that may have lasted only minutes. (In general, the more detailed the model, the greater the number of input parameters it requires, and the greater the accuracy that is required of these. Hence, it is not merely a question of providing faster computers.)
- *Model applicability*: since these models have not been developed with emergency response in mind, their outputs will usually comprise the sort of information that fire modellers are interested in – which is not necessarily that most relevant to the incident commander who is trying to assess the situation and determine an appropriate course of action. Furthermore, even where they are useful the outputs may contain (or express)

some degree of uncertainty, which again is potentially problematic for the incident commander who would prefer to deal in certainties.

While the first of these issues remains, the FireGrid approach is intended, in some measure, to address the latter two. Access to HPC resources would allow models to be executed quickly enough that their results are made available while still relevant to the emergency response. One means of accessing these resources (and in a uniform manner) is across the Grid, thus also introducing the element of Grid computing into the project. However, the use of HPC/Grid raises another issue, since it cannot be assumed that existing models will have been developed to use such resources in the most optimal fashion. The use of 'live' sensor data from the location of the emergency would go some way to overcoming the problem of the model requiring detailed inputs, and might be used in an interactive manner to 'steer' the simulation towards more accurate results. (However, the available models typically do not behave in this 'sensor-steered' way.) Since potentially many models might require access to the data, and it might have a valuable role to play in post-incident analyses, a database would be required to store sensor readings. Finally, AI approaches to knowledge modelling, mapping and presentation could help to represent the outputs of the models in an appropriate form, and situating the whole in (from the perspective of the user) an agent-based framework would allow the system to be constructed and deployed in a modular fashion.

3 A Grid-Enabled Agent-based Infrastructure for Emergency Support

Having described the rationale behind the FireGrid approach, and the elements of a system that this approach entails, in this section we describe in detail how we use these different technologies to construct an infrastructure for supporting emergency response. The FireGrid architecture consists of four 'layers': a data acquisition and storage layer for capturing and storing live sensor data; an HPC resource layer for deploying computational models; a Grid middleware layer for accessing HPC resources; and an agent-based command-and-control layer to allow fire responders to interact with data, computing resources and the Grid, in order to perform high-level decision-making. Figure 1 shows the interaction of these different layers, each of which we will now describe.

3.1 Data Acquisition and Storage Layer

Within the FireGrid system, a central data repository has been established for storing all related information generated in the system. The repository stores both *dynamic* data and *static* data. Dynamic data is the information that is continuously fed into the database from the sensor network. Different sensors such as smoke detectors, thermocouples (temperature sensors), CO and CO_2 detectors, etc., are deployed in the building for continuous monitoring of its state. In the event of fire, the output from each of these sensors provides potentially valuable data for input to computational models. Since sensor data can be noisy or can fail due to manufacturing flaws or the extreme conditions of the fire incident itself, the data must be filtered to provide some measure of the accuracy and quality of data received from the building. These real-time sensor data are then stored in the system database. In addition to this dynamic data, the database holds static data that is (reasonably) time-invariant data such as the geometry/layout of the building, the types of sensors and their locations, the types and material of furniture and the location of fire suppression systems. Apart from physical and material properties, static data may also include pre-computed scenarios of fire development, which can be compared with the real fire state so as to steer computational models.

3.2 HPC Resource Layer

This layer consists of appropriate simulation models deployed on specific HPC resources (and optimised to run most effectively on those resources). In the ideal case, the models - would be independent of any particular context (such as a specific building) or incident and would rely for its initial conditions on information acquired from the system database, along with updates based on data sensor whenever appropriate. However, as mentioned above in section 2 existing models cannot be assumed to meet these requirements without at least some re-engineering effort, which, on a practical level, limits the models available. (For the purposes of this paper we concentrate on simulation models which require the computational power of HPC resources; however, for models that do not require this power – for example, models that provide a simple interpretation of current data – there is no reason why these should not be run on any convenient computer resource, and indeed, given the overheads of interacting with models on HPC resources, this is usually a better alternative.)

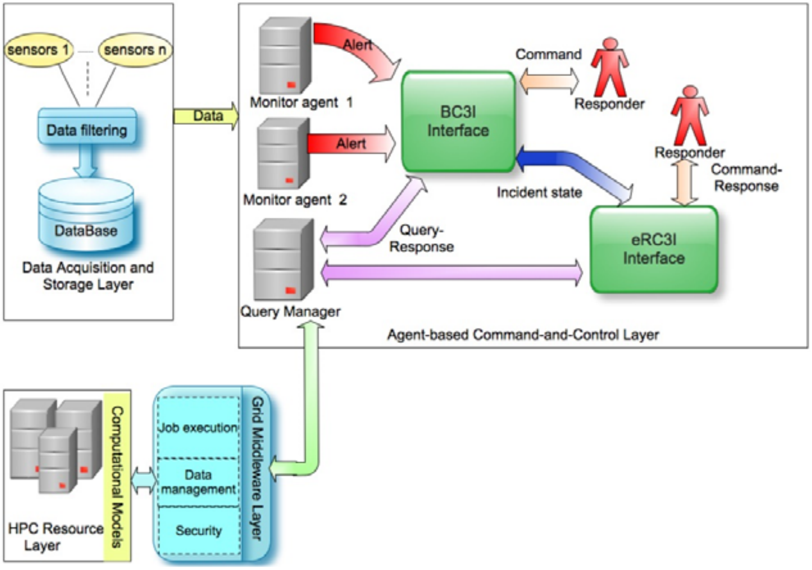


Figure 1. The FireGrid architecture for emergency response support.

3.3 Grid Middleware Layer

A Grid is a geographically distributed computation platform that can allow users to access various (High Performance and otherwise) computing resources, via a uniform computational interface. A FireGrid system would require not only access to computing resources when required but also the management of distributed data resources, which is a combination typical of Grid applications, involving both on-demand computing and data-intensive computing [9]. Since the system is intended to be used during possibly rapidly developing emergencies, near-immediate access to computational resources is required. In practical terms, this requires that a compute job emanating from a FireGrid system is treated as a priority by the middleware management layer controlling the HPC resource. In summary, functional requirements are the provision of a job execution service for running models on remote resources, the staging of input files for the models to the remote host, the transfer of output files from the remote host back to the client after job completion, the monitoring of job status, and the provision of security and authorisation services.

3.4 Agent-Based Command-and-Control Layer

Users of a FireGrid system need to be able to access the information generated by the combination of data, HPC and Grid components and, in certain cases, to be able to formulate requests for specific information (to be fulfilled, where possible, using the outputs of particular models). This interaction with a FireGrid system is the role of a command-and-control layer. This layer requires a certain amount of configurability in order to tailor it to the needs and resources of particular contexts (i.e., buildings and their available sensor systems) and users (whose involvement may depend on the nature and current state of the incident). We also require that the system presents, as far as possible, a seamless and consistent face to its users. Consideration of these requirements has led us to adopt an agent-based model for the command-and-control layer in which both human and computer agents collaborate.

An agent is some ‘intelligent’ entity that exists within some environment alongside other agents with which it can communicate and interact [10]. Agents typically differ in their knowledge and capabilities, and may have access to different data. A multi-agent system [11] is a collective of agents that work cooperatively to achieve common goals that cannot be achieved by any of the agents acting individually. In order to achieve goals coherently, the agents must communicate with each other and coordinate their activities. In the FireGrid command-and-control layer, we can identify three basic types of agent:

- One or more *Command, Control, Communication and Intelligence* (C3I) agents. Each of these is composed of a human in a decision-making role and an interface (with underlying reasoning mechanisms) conveying the state of the building in question as interpreted by a FireGrid system. The nature of the interface (and its reasoning) is determined by the nature of its intended user. A user might be, for instance, a fire officer in a commanding role dealing with a full-scale incident, or a member of the local security staff who might be responsible for monitoring the building, and instigating evacuation and calling out the fire brigade if a fire is detected, but who would not be expected to tackle personally anything but the smallest fires.
- One or more autonomous *system monitor* agents, used to monitor the status of environments (such as a fire alarm agent that monitors sensor data for evidence of a fire and which can alert the user accordingly). The number and nature of such agents will depend on the environment in question, and the location and types of sensors present.
- A *query manager* agent. This agent has two main tasks: answering queries for specific information that are requested by users (or in some cases their interfaces will make these requests autonomously on behalf of its user) through interacting with the available data-interpreting models, and arranging for resources to be scheduled and managing data for these models through interacting with the Grid middleware layer. When receiving a query from a C3I agent, the query manager will search for any available computational model that is capable of answering this query. If such a model is found then the query manager will interact with Grid middleware layer to invoke the model, produce an answer to the query and then pass it back to the requester. If the model is not found, the query manager replies with a message to this effect and awaits the next query.

In addition, there may be a number of humans (e.g., fire-fighters) and autonomous or semi-autonomous units (e.g., sprinkler systems) that might validly be considered part of the command-and-control layer, but we shall consider these no further here.

It is through the query manager agent, then, that the user accesses the interpretations of the sensor data produced by computational models running on HPC resources. In order to formalise a query for information, a ‘language’ has been developed based on notions of physical states and events [7]. An incident state parameter is some measurable quantity at some location, such as maximum temperature, that persists for some duration of time during the incident, which is measurable throughout that duration, and which is expressed according to some convention. An event is some phenomenon that is assumed to occur instantaneously at some location during the incident (such as an explosion or collapse); in other words it has no temporal extension, but instead occurs at a point in time. Queries represent requests for the values or locations in time or space of some state parameter or event, while the same language is also used for specifying the types of request that a particular model can answer, enabling queries to be matched to appropriate models. While this language gives a basis for agent communication, knowledge engineering is required to ensure that, for instance, requests are converted into appropriate invocations of models, that output files are interpreted and expressed correctly in terms of the language, and that the content of responses is presented appropriately to users via their C3I interfaces. The effort required for this can be significant, and requires agreement between modellers, users and system builders.

4 An Application Case Study

For reasons that should be obvious, validating the FireGrid architecture presents a number of practical difficulties. The project has adopted an incremental approach, gradually increasing the numbers of implemented components, and based on simulated, pre-recorded and live data collected from fires. As a result of this process, a number of different FireGrid systems have been implemented that conform in part or in whole to the architecture described above, and culminating in a 'complete' system that was applied to provide real-time information during an experiment involving a real fire (under controlled conditions) that was conducted before a select audience at the Building Research Establishment (BRE), near London, in October 2008. In this section we shall describe this experiment and its results.

4.1 Experimental Rig

The fire was to occur in an 'apartment' (constructed in the Burn Hall at BRE) consisting of three rooms connected in a T-shape plan by a corridor. Each room was a cube of side 2.4m. The rooms were connected by a corridor 3.6m in length and 1.2m wide. Room 1 was to be the location of the source of the fire. It would also contain typical household furniture (sofa, table, television, bookshelves), all potential fuel for a fire. A technician would start the fire by igniting the sofa. There was to be a total of 125 sensors located throughout the rig; these included: smoke and fire alarms; thermocouples to monitor surface and air temperatures; radiometers to measure heat flux; gas (O_2, CO, CO_2) sensors; and deflection meters to monitor deformation of structural elements. Values from each of these sensors would be polled in batch mode at roughly 3-second intervals.

This rig and its contents were intended to produce a 'flashed-over' fire. Flashover typically occurs when the gases produced by a fire in some enclosed space reach temperatures high enough to ignite simultaneously all combustible matter in the vicinity. The conditions required for flashover are controversial among the fire engineering community, but it usually occurs at temperatures at or above 500°C and, from the perspective of responders, represents a potential transition from a contained fire to an uncontrolled fire. In addition, certain structural elements of the rig were expected to deform and fail during the fire; the potential collapse of ceilings and floors is, of course, a major hazard for fire-fighters.

4.2 FireGrid System

The system was developed with a particular end-user in mind, namely (using UK fire service terminology) a Fire Incident Commander, or – more likely – a support officer detailed to assist the Incident Commander [12]. The Incident Commander is responsible for the management of the incident, including tactical planning, coordination and resource deployment. In this experiment, the Commander's decision would be whether or not to send officers into the rig to search for occupants trapped by the fire or overcome by fumes; however, there would be no direct intervention in the actual fire.

4.2.1 Data Acquisition and Storage Layer

A PostgreSQL database (running on a server physically located at the University of Edinburgh) was used to store both the static and the dynamic data from the experiment. The static data was entered manually into the database before the experiment and included the dimensions of the rig, and the locations and types of each of the sensors. The dynamic data was to consist of the sensor readings measured during the course of the experiment. This data was to be sampled in raw form (voltages) by a data logger, a specialised hardware device connected to a PC; in software the readings would then be converted into the appropriate values and calibrated. Finally, a simple filtering algorithm would be applied to each data value to determine those that lay outside the expected operating range for the type of sensor in question; these values were flagged as invalid, and all values were then to be time-stamped and written as a batch to the database over an internet connection.

4.2.2 HPC Resource Layer

The principal model selected for simulating the fire was K-CRISP [13][14]. This model is a sensor-linked zone model that has been developed by members of the FireGrid project (an important point, since this allowed ready access to source code for compilation on the HPC resource and for any necessary functional modifications). The model adopts a Monte Carlo approach, with multiple alternative scenarios for the development of the fire after ignition time generated from a set of initial conditions, and as such it readily lends itself to parallelisation, and maximum exploitation of however many processing cores are available at the HPC resource. The model would be supplied with real-time sensor readings, which would be used to determine the current best scenarios (that is, those found to most closely match the readings seen thus far). As well as giving some indication of the predicted evolution of the fire, the best scenarios are also used to implement a feedback loop to modify the parametric space from which initial conditions for new scenarios are drawn so as to gradually steer the simulation towards the real fire conditions. The K-CRISP code is written in FORTRAN, with input data and output results in the form of text files. While K-CRISP can run on a PC, its Monte Carlo approach means that generating more scenarios increases the likelihood of generating better scenarios, and hence executing the code on a HPC machine should have the effect of improving the quality of its results.

For this FireGrid system, based on the content of the current best scenarios, K-CRISP output would be interpreted to provide predictions for the values of different state parameters (maximum temperatures, smoke layers, visibility, etc) in each of the rooms at two-minute intervals into the future, as well predictions of any flashover or collapse events. Two HPC resources were available for the experiment, namely the Edinburgh Compute and Data Facility (ECDF) at the University of Edinburgh and HPCx, the UK National Academic Supercomputer, which was to be a backup machine in the event of failure (in the event the HPCx resource was not needed). Differences in processing architectures and operating systems meant that the K-CRISP code had to be compiled separately for each resource.

4.2.3 Grid Middleware Layer

The role of the Grid middleware in this FireGrid system, then, is to provide access to K-CRISP on the HPC resources. On the client side, jobs would be submitted via the query manager agent. This would be done using the CoG Kit [15], integrated into the query manager code, to submit jobs and file-transfer requests to the Globus Tookit v4 (GT4) [16] installed on the HPC server. GT4 would then submit the jobs to the batch-queuing system on the resource in question, and mediate in the transfer of files and monitoring information back to the query manager client. Since Grid access to the available HPC resources is dependent on certificate-based user authorisation, a member of the FireGrid team had to register his certificate details with the resource managers, and the query manager would effectively be run in the name of this user. GT4 also provided the required run-time authorisation.

One requirement of FireGrid is that, since it is to be used in emergency situations, any jobs should be treated as urgent and scheduled to run as soon as possible. However, even with the highest priority attached to the jobs, it was found that the implementation of the queuing systems is such that it might take up to two minutes for a job to begin executing. To overcome this unacceptable delay, methods of ensuring immediate access to reserved processors were requested (and provided) on each of the HPC resources. Finally, since the number of available nodes on the two HPC resources differed, it was necessary to write a different executable script to provide optimal allocation of K-CRISP jobs for each node.

4.2.4 Agent-Based Command-and-Control Layer

As mentioned in section 3.4, a number of different agents are required to implement the command-and-control layer. Each of these agents was developed using the I-X approach and software suite [17], developed by one of the project partners, which provides a generic framework for the support of processes among collaborating human and computer agents.

The query manager agent would provide the link between this layer and the HPC resources. At this level the K-CRISP model had to be ‘wrapped’ with code that would retrieve the latest sensor readings from the database and construct the appropriate input files, and then process its output files in order to extract predictions and compose these into the appropriate messages for the C3I interface. This latter step proved difficult since the output of the model as a whole is probabilistic yet the fire-fighters want deterministic information, and finding a satisfactory compromise to this mismatch was not straightforward. In addition, local, simple (that is, non-HPC) models able to determine the current maximum temperature and gas concentrations in each room were written and incorporated into the query manager.

A fire alarm monitor agent was also provided to monitor the database for indications of a fire, and then to send an appropriate alert to any C3I agents. Since in this case the sensors already included smoke and fire alarms, this proved to be a relatively trivial agent.

For this experiment it was decided to have a single C3I agent, corresponding to (the officer supporting) the Incident Commander who would use an I-X interface. With the role of the Incident Commander in mind, and with the involvement of serving officers from the London and Lothian and Borders Fire Brigades, a graphical interface was developed which aims to convey succinctly and rapidly to its user the current ‘hazard level’ at each location (in this case, each room) within the building. This hazard level expresses an integrated measure of the degree of the risk to which a fire-fighter operating at that location would be exposed. The hazard level is represented using two ‘traffic lights’, one on top of the other, for each room, where a green light should be interpreted as “the system is unaware of any specific hazard to fire-fighters operating under normal safe systems of work at this location”, amber as “additional control measures may need to be deployed to manage hazards at this location” and red as “this location may be dangerous for fire-fighters”. The lower traffic light indicates the current hazard level in the location, whereas the upper light indicates the ‘worst’ predicted hazard level for that location (which is never better than the current level). The user is able to select any location to get further details about its state, including the nature of any detected hazards, and to explore any predictions by viewing the projected state at different times in the future (this interface is shown in Figure 2).

Underlying this is a reasoning mechanism in the interface that maintains a set of ‘beliefs’ about the values of state parameters and occurrence of events. These beliefs are derived from the messages sent, via the query manager, from the models. Given the temporal content of these messages, and the need to resolve conflicts (usually by favouring more recent information about current state) the reasoning involved in the maintenance of the belief set is quite complex. A set of ‘hazard rules’ is then applied to the belief set to derive the hazard level at any particular time (for instance, a simple rule might state that if flashover occurs in some location then the hazard level of that location is red from that time onwards). These rules could be tailored to the needs of specific deployments of FireGrid systems and specific users (with whom, indeed, the rules ought to be developed).

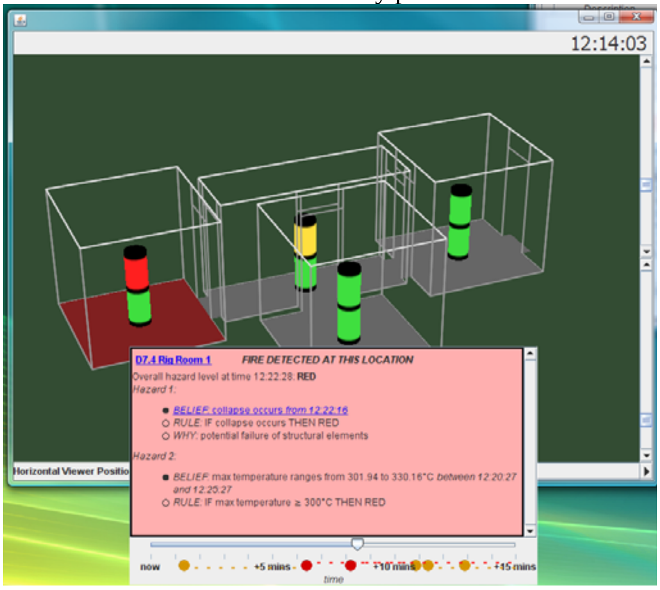
Finally, it was decided that, on receipt of a fire alarm the C3I interface should automatically send the query that would launch K-CRISP. The user could invoke the other simple models by constructing appropriate queries using simple drop-down menus.

4.3 Results

With a member of the FireGrid team playing the role of support officer to the Incident Commander (but with a senior fire officer in the audience), the experiment was conducted and was felt, in general terms, to be a success, with the system behaving as envisaged. The fire, on the other hand, did not behave entirely as planned (which serves to underline the difficulty of making predictions about the course of fires!): rather than rising directly to flashover, temperatures instead rose slowly and initially peaked at a sub-flashover level, remained at around this value for some time before decreasing, but then (probably provoked by the spread of the fire to a table) rapidly flashing over, leading to high temperatures and smoke throughout the rig, and the failure of structural elements. The fire burned for approximately one hour before it was manually extinguished.

Throughout the experiment, data was streamed to the database from the sensors in the rig; following the detection of the fire and the submission of appropriate queries to the query manager, this data was then used by simple models to provide both 'live' updates of the current state of the incident, and by K-CRISP to provide predictions of the future development of the incident. These results were relayed to the user (and the audience) through the C3I interface. Figure 2 shows the C3I interface some time into the experiment displaying predictions from K-CRISP.

However, while the infrastructure did work as intended, the predictions generated by the K-CRISP model were not particularly accurate – predictions of flashover and collapse (although later retracted) assumed a continuous increase to flashover temperatures, and the actual flashover was not predicted at all (with K-CRISP at that point still forecasting a decrease in temperatures). In large part this seems due to the unexpected behaviour of the fire, which was more complex than K-CRISP can accurately simulate, and some blame might also be attached to the approximations and simplifications introduced in order to interpret model results. Ultimately, though, the behaviour of fire is not wholly understood, with the number of potentially influential factors being very great, and it seems unlikely that any currently available model could have accurately predicted the course of this fire.



D7.4 Rig Room 1 FIRE DETECTED AT THIS LOCATION

Overall hazard level at time 12:22:28: RED

Hazard 1:

- *BELIEF:* collapse occurs from 12:22:16
  - ○ *RULE:* IF collapse occurs THEN RED
  - ○ *WHY:* potential failure of structural elements

Hazard 2:

- *BELIEF:* max temperature ranges from 301.94 to 330.16°C *between 12:20:27 and 12:25:27*
  - ○ *RULE:* IF max temperature ≥ 300°C THEN RED

Figure 2. C3I interface around 15 mins after detection of the fire (the red floor shows the location of the fire). The user has clicked on this room to see further details. While the current hazard level of this room (and of all other locations) is green, two red hazards (derived from a projected collapse and lethal temperatures) and four amber hazards are predicted in the next 15 mins.

5 Discussion

It should be clear that FireGrid systems, including predictive capabilities, are not going to be deployed as real emergency response aids any time in the near future. The aims of the FireGrid project place the emphasis on the integration of existing technologies into a functional prototype system that illustrates the potential advantage of the approach, and in this respect it is felt that the project has been a success. Even without dependable predictions, the experimental system, with its flow of 'live' data from sensors to database to interpreted presentation at the C3I, was felt to provide fire-fighters (and fire modellers) with a new and privileged insight into the conditions within buildings during fires, and one which would have influenced intervention in the fire. However, as is often the case with such speculative, inter-disciplinary projects, it has also revealed a number of shortcomings in the approach.

Modelling technology is some way from being able to predict the evolution of fires with accuracy and precision required for emergency response; furthermore, since models have not been developed with this end in mind, their outputs require a degree of interpretation that they do not always comfortably accommodate.

A related question involves the number of sensors; fire models are highly parameterised, and in order to fix as many of these parameters as possible, a large number and type of sensors were installed in the experimental rig, and some of their locations (such as those supported in the middle of rooms to measure air temperatures) would present an annoyance to anyone living or working in that space. At this degree of instrumentation, FireGrid seems to appeal to only high-cost or high-risk structures – and that may continue to be the case, although advances in sensor technology (and decreases in cost) may widen its applicability. Further problems are that of identifying failed or failing sensors, and of ensuring that the system degrades gracefully in the face of those failures anticipated to occur during a real fire.

The HPC/Grid element, while ostensibly a compelling route to faster simulation, introduces a new set of problems. Unless developed with HPC deployment in mind, existing code is unlikely to exploit fully the potential, and given the overheads involved, HPC deployment may actually lead to worse performance. It may be that, given the shortcomings of the models, a better strategy would be to try to focus efforts on approaches (either new ways of building new models or new ways of using existing models) that require less computational power yet still provide useful results.

The knowledge modelling underpinning the C3I layer could, if not done with care, lead to dangerous misinterpretations of sensor data and model outputs. There is a tension between, on the one hand, providing the concise, authoritative information responders demand and, on the other, respecting the complexity and subtleties of model results. This leads to consideration of the operational validation of the approach and any systems based upon it, typically a problem for AI applications which, by their nature, tend to deal with heuristic and approximate methods, rather than certainties and guaranteed results. Here, of course, system validation is compounded by the fact that large-scale emergency incidents can, at best, only be simulated under laboratory conditions, and then at considerable cost.

Notwithstanding these difficulties, the potential that has been demonstrated is exciting enough that the partners confidently envisage that development will continue, ultimately leading to the deployment of systems based on FireGrid concepts; moreover, we expect the full backing of interested stakeholders – fire-fighters and emergency responders more generally, architects and construction firms, insurance companies – in these endeavours.

6 Conclusions

Emergency response is an important concern in modern societies; it requires the urgent and coordinated invocation of resources including human responders, organisations and relevant services in a timely and effective manner. Technologies like those mentioned in this paper could come to play an effective role in any response – and, indeed, they have the potential to revolutionise the way in which responders approach their task.

Through an imaginative integration of advanced sensor, computational model, Grid and multi-agent system technologies, the FireGrid project proposes a novel e-Infrastructure for next-generation emergency response support: live sensor data captures the unfolding situation; the Grid enables uniform and secure access to distributed HPC resources; models running on these resources provide interpretations of the current and projected future states of the incident; and an agent layer seamlessly delivers these interpretations to the users of the system as decision-making intelligence. The results of experiments suggest that, while some areas are more developed than others and difficulties undoubtedly remain to be overcome, the approach is extremely promising.

7 Acknowledgments

The work reported in this paper has formed part of the FireGrid project. This project is co-funded by the UK Technology Strategy Board's Collaborative Research and Development programme, following an open competition. The authors acknowledge the contribution of all colleagues in the FireGrid project. This work made use of the facilities of HPCx, the UK's national high-performance computing service, which is provided by EPCC at the University of Edinburgh and by STFC Daresbury Laboratory, and funded by the Department for Innovation, Universities and Skills through EPSRC's High End Computing Programme. The University of Edinburgh, project partners and project funding agencies are authorized to reproduce and distribute reprints and on-line copies for their purposes notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of other parties.

References

[1]. Berry, D., Usmani, A., Torero, J., Tate, A., McLaughlin, S., Potter, S., Trew, A., Baxter, R., Bull, M. and Atkinson, M. (2005) FireGrid: Integrated Emergency Response and Fire Safety Engineering for the Future Built Environment, *UK e-Science Programme All Hands Meeting (AHM-2005)*, Nottingham, UK, Sept. 19-22. See also http://www.firegrid.org/.

[2]. Fuhrmann, S., Brewer, I., MacEachren, A., Cai, G., Sharma, R. and Wang, H. et al. (2003) Collaborative emergency management with multimodal GIS. *In the ESRI International User Conference*, San Diego, CA.

[3]. Gheorghe, A. V. and Vamanu, D. V., (2001) Adapting to new challenges: IIDS for emergency preparedness and management. *Int. Journal Risk Assessment and Management*, 2(3/4):211–223.

[4]. Tufekci, S. (1995) An integrated emergency management decision support system for hurricane emergencies. *Safety Science*, 20(1):39–48.

[5]. Hernandez, J. Z., and Serrano, J. M. (2001) Knowledge-based models for emergency management systems. *Expert Systems with Applications*, 20(2):173–186

[6]. Schoenharl, T. and Madey, G. (2006) WIPER: A Multi-Agent System for Emergency Response, *Proc. 3rd Int. Conf. on Information Systems for Crisis Response and Management (ISCRAM 2006)*, B. Van de Walle and M. Turoff (eds.), Newark, NJ, USA.

[7]. Potter, S. and Wickler G. (2008) Model-Based Query Systems for Emergency Response, *Proc. 5th Int. Conf. on Information Systems for Crisis Response and Management (ISCRAM 2008)*, F. Fiedrich and B. Van de Walle (eds.), Washington DC, USA, May 2008.

[8]. Cowlard A., Jahn W., Empis CA., Rein, G., and Torero JL. (2009) Sensor Assisted Fire Fighting, *Fire Technology* (in press), 2009.

[9]. Foster, I. and Kesselman, C. (1999) *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann.

[10]. Pissinou, N., Kurkovsky, S., Benton, R. and Bhagyavati, B. (1997) A roadmap to the utilization of intelligent information agents. *In Proc. IEEE Knowledge & Data Engineering Exchange Workshop, KDEX97*.

[11]. Wooldridge, M. (2002) An Introduction to Multi-Agent Systems. *John Wiley and Sons Ltd*.

[12]. HM Fire Service Inspectorate (2002) Fire Service Manual, Volume 2 Fire Service Operations, Incident Command, HM Fire Services Inspectorate Publications, London: The Stationary Office.

[13]. Fraser-Mitchell, J. N. (1994) An object-oriented simulation (CRISP II) for fire risk assessment. *Fire Safety Science,* 4:793–804.

[14]. Koo, S.K., Fraser-Mitchell, J., Upadhyay, R. and Welch, S. (2008). Sensor-linked fire simulation using a Monte-Carlo approach, Proc. 9th Int. Symp. Fire Safety Science, Karlsruhe, Germany.

[15]. von Laszewski G., Foster, I., Gawor J., and Lane P. (2001) A Java Commodity Grid Kit, Concurrency and Computation: Practice and Experience, 13(89):643-662, 2001.

[16]. Foster, I. (2005) Globus Toolkit Version 4: Software for Service-Oriented Systems, *IFIP Int. Conf. Network and Parallel Computing,* Springer-Verlag LNCS 3779, pp 2-13, 2005.

[17]. Potter, S., Tate, A. and Wickler, G. (2006) Using I-X panels as intelligent to-do lists for agent coordination in emergency response. *In Proc. 3rd Int Conf on Information Systems for Crisis Response and Management (ISCRAM 2006)*, pp. 272–281, Newark, NJ, USA.