**Abstract** - Developing computer technology provides to use the information system commonly. All to gather, big data processing is growing importance. Moreover to operate distributed and dynamic environments are inevitable in real life to realize process in a short time. Decision-making processes among autonomous agents can support to solve dynamic and large system problems. This paper presents an agent based emergency response coordination model that considers the graph data model and associated query language provides a unifying conceptual model. We used the Neo4j which is Cypher graph query language.

**Keywords:** Graph databases, Source-code queries, Multi agent system, Emergency Response Model

I. **INTRODUCTION**

Graph database uses the graph structures with nodes, edges and properties to represent and store data. It is a storage system that provides the index-free adjacency. General graph databases that can store any graph are distinct from specialized graph databases such as triple stores and network databases. Edges are the lines that connect nodes to nodes or nodes to properties and they represent the relationship between the two. Most of the important information is really stored in the edges (Neo4j, Mens; 2005, 2007). Meaningful patterns emerge when one examines the connections and interconnections of nodes, properties, and edges. In this study, emergency response model designed with graph database approach.

Emergency response model covers the nature events and the non-nature events. In particular, unexpected events cause the turmoil among the people most of time. To prevent the turmoil and to save their life in every time, in particular depend on the emergency responses. Appropriate responses are needed in the form of allocating resources to handle the effects of emergency responses. The form and remedy kind may help to people easily kind of the unexpected event.

Also the early caution systems can able to stimulate people to this kind of events.

Determination of the unexpected events type is so important for help the people. Every time, people faces to nature events such earthquake, forest fires, terrorist attacks, war threads. Also to operate distributed and dynamic unexpected environments are inevitable and so important in real life to realize process in a short time. Multi-agent emergency response system has been extensively used in the different tasks of decentralized emergency response problem solving such as communication among agents, collective decision making, cooperation, collaborative planning in large scale that deals with uncertainty and conflicting information during emergency response management.

The paper is organized as follows: the next section presents an overview of works in the literature related to graph database approach, emergency response structure, multi agent systems dynamic modeling; the next section introduces overall architecture which attains the proposed aims and emphasizes the roles played by the agent based emergency system components; and describes the complete working flow and details theoretical approach on which relies the work; then, the next describes the process model applied to emergency response model case study. Conclusions and future works close the paper.

II. **LITERATURE SURVEY**

The graph data model unifies queries cross-cutting over various representations of source code, also demonstrate a prototype implementation based on Neo4j. It stores full source-code information and scales. The graph data model encodes entities and relationships amongst them using a directed graph structure (Angles, Gutierrez, 2008). It consists of sets of nodes and binary edges (and hyperedges but we do not use these here) representing entities and relationships between these entities. Nodes and edges can be labelled, typically with the name of the entity.
they represent, but possibly with a tuple of name and further values.

Graph databases provide to combine relations based on a common attribute to retrieve connected records. It is faster and this potential source of inefficiency that is addressed in some graph databases which provide index facilities to map certain structures known ahead of time to a list of records (Vicknair et al. 2010). In addition, because graph databases do not depend on a schema, they can be more flexible when the structure of records needs to be changed.

Emergency management is so important for the sustainable qualify life conditions. In particular, to take precautions for unexpected emergency situations are so important for the human life. MAS may use in emergency situations resulting from natural and human made emergency responses, such as flood, tsunami, earthquake, terrorist attack, fire in building etc, represent complex and dynamic environments with high level of uncertainty; hence autonomous notification and situation reporting for emergency response management system will be done by multi-agent response system. In literature, Wang et al. (2013) suggested the emergency management response system structure for a city in China. Also, Basak (2011) et al., suggested the agent based disaster management and Chou et al. 2008, suggested the dynamic parking structure with agent based platform. Belief Desire Intelligence (BDI) is very important architecture for offer an agent with artificial capabilities. Some of the emergency response response systems have been developed based on multi-agents systems approach (such as: DrillSim developed by Balasubramanian et al. 2006, DEFACTO designed by Marecki et al. 2005, ALADDIN modeled by Adams et al. 2008 and Jennings et al., RoboCup Rescue suggested by Kleiner et al. 2005, and FireGrid proposed by Berry and Usmani in 2005) and more are being developed.

III. AGENT BASED EMERGENCY MODEL

This section includes overall architecture which achieves the proposed aims and emphasizes the roles played by all system components; and describes the complete working flow and details. Theoretical approach on which relies the work. The next subtitle of the Agent Based Model gives the detail of the emergency responsive model with components.

3.1 The Cycle of the Agent Model

Our suggested model could be done - situation assessment; understanding context surrounding; communication; collaboration properties. The suggested model is able to realize communication and collaboration with coordination, coalition and collaborative information distribution properties. Agents communicate in order to achieve better the tasks of them or of the society/system in which they exist. Communication can enable the agents to coordinate their actions and behavior, resulting in systems that are more balanced. For communication, agents are able to coordinate, coalition and collaborative information each other agents.

The autonomy is an ability of agent to achieve its goals without any supporting from other agents. On the other hand, the interaction of agents to get the global goal of the system is the social ability of agents. The reactivity, which is based on the relation between perception and action, is an ability of agents to respond to the environmental changes. The pro-activeness of agents is an ability to express the goal-directed behaviors. The reactions of agents to the environmental changes are the reactivity or pro-activeness that depends on what kind of architecture of agent is used to develop agents. Intelligence is the ability of the agent using its knowledge and reasoning mechanisms to make a suitable decision with respect to the environmental changes.

Figure1. Dynamic modeling of the abstract system
3.2 Emergency Response Agent Based Model

In this paper, the current emergency response management and response systems are analyzed and taken into consideration of domain requirements, agent-based design methodology and systems’ comparative view. Suggested agent model covers the some parts, which are collective information, coordination, communication, coalition, cooperative distributed information sharing, problem solving and evaluation, decision making. Contract Net (CNET) is preferred to achieving efficient cooperation through task sharing in networks (Weiss, 2012) as well as used in Emergency response System for communicating problem solvers.

Multi-agent emergency response system includes the different tasks of distributed emergency response problem solving such as collective decision making, communication among agents, cooperation, collaborative planning in large scale that deals with uncertainty and conflicting information during emergency response management (Adams et al.). In detail, the suggested type of emergency response systems can be viewed on information and knowledge fusion and take the feedback from the existing agents for sensing, coordinating, decision making and acting. It must be able to achieve these objectives in environments in which: control is distributed; uncertainty, ambiguity, imprecision; multiple agents with different aims and objectives are present; and resources are limited and vary during the system’s operation.

The suggested model may provide some benefits which are); more robust, interoperable, and priority sensitive communications, better situational awareness, improved decision support and resource tracking, greater organizational agility, better engagement of the public. Emergency response model covers the very wide area respects of the nature events (earthquake, floats, fires, hurricane, and thunder storms), terror events (considering of the coming information), fault of the people or devices (accidents, fires, explosions etc.).

**Knowledge Base Agent:** It includes the natural events, terror events and accidents, also natural events include the earthquake, flood, fire, hurricane, statistical data and current data. The knowledge base agent takes data from environment observation. Also system inputs connected with the knowledge base with simultaneously coming data information, historic data and weather data. At the same time, learning process shares the data with system inputs and knowledge base. It covers the database, which is related to the system components. It provides the using of the information whole system.

In order to deal with a lot of data, connection between stored data and making decisions on collected data, a graph, schema less database has been selected. In graph databases, every element contains a direct pointer to its adjacent elements and no index lookups necessary. It allows analyzing the data fast. Having schema less structure helps to store a lot of useful data on the nodes. Storing data this way allows making suggestions for action plan. In knowledge based agent, the applied area is stored as City, County, District, Street. These areas can be assumed as nodes and the connection between them locates the hierarch, as mentioned from right to left. Cities will also be connected to each other so the model can be completed and therefore search will be easier in the system. The applied area and event type modeled in Figure 2 and 3.

![Figure 2 Main structure of the location](image)

Thus the system will be able to store an event by address. Also events types will be nodes such as

![Figure 3 Main structure of the event types](image)

When an event occurs or gets reported, the system will take information and store the event according to the given address. Figure 4 shows how it gets stored.

![Figure 4. Taking information and store events diagram](image)
The events are connected to the event types and also has a response action. From an event type, events in different cities can be reached and response actions can also be recommended for another event. Figure 5 shows the event and event type structure.

Figure 5 Event and event type structure

By storing the data in this structure, we are able to find events occurred in a street, district, county, city, what kind of actions has been taken, information about those taken actions. We can also suggest response actions for event according to its type as event types connected to different events and those events might have similar response actions. This structure will allow the system travel across the country easier and faster. By mean faster, the response will be quicker to resolve the event.

**Performance Estimation Agent:** It includes the information and coming data information. Processing of the information provides the coming data analysis and evaluation with proper decision making structure.

**Service Provider Agent:** The agent includes the service system which includes the ambulance service, logistic service, police linked service, combined emergency service, hospital based service.

**Decision Making Agent:** It includes the system outputs which are prediction of the event type, prediction of the location, prediction of the correct resource usage and correct resource coordination.

The system goal in this agent-based coordination network between emergency support system and people who are needs to support is divided into three parts: defining the initial negotiation strategy, considering the emergency assessment factors, and finding the cost functions and report results.

The architecture of an agent shown in Figure 6 consists of five modules: perception, decision making, knowledge, control, and communication. The perception module is responsible for data acquisition in the environment. The decision making module is in charge of the agent making a decision in an autonomous way.

![Figure 6 Agent Based Emergency Response Model](image)

The control module processes the plan into tasks and executes the tasks to the environment. At the same time, this module sends the tasks to the communication module if the plan is processed by the agent team. The communication module is responsible for interactions between the agent and the agent community. It receives or sends messages, interprets them and transmits the tasks of the control module to other agents. The knowledge module contains intentions, and plans of the agent.

Control module is so important for the decision making agent. It covers the update, process of the task, evaluate of the alternatives, compare, learning, announce, negotiate, notify, request, notification and computing. The system considers the Table 2 control activities.

| B | Update (B, s) |
| D | Process(task) |
| E | Evaluate (alternatives) |
| T | Compare (B, D, E) |
| L | Learning (situation) |
| A | Announce |
| N | Negotiate |
| N | Notify |
| R | Request notification |
| C | Computing |
Table 2 Summary of the activity list in control module

The case study covers the some steps which are system analysis and design; building of conversation; system processes.

3.3 System analysis and design

The coordination network covers the system component, experts and people. FIPA –Contract-Net-Manager protocol is applied to analyze and design the system. Finally, the pattern of this system is built and illustrated. This study emphasizes the coordination of the network emergency assist system and who needs the assist one. The agent mechanism structure can be applied to establish such a negotiation system with the distinction from the emergency information system, and at the same time based on research as shown in the literature review.

3.3.1 System analysis

The agent can complete complicated negotiation with related agent. Agents work automatically without rest on coordination, enabling the drivers to seek out. As the agent architecture, the FIPA – Contract-Net-Manager protocol is preferred.

3.3.2 Agent classes

In negotiation, the agents engage in dialogue, exchanging proposals with each other, evaluating other agents’ proposals and modifying their own proposals and then modifying their own proposals until all agents are satisfied with the set of proposals. Standard negotiation mechanisms adopted are based on game theory or on human-inspired negotiations. Every task also defines higher level, complex interaction protocols requiring coordination between multiple agents. First, the system checks emergency assist opportunities information for user when he inputs data into the system, and sends a message to emergency support agent. Second the user agent negotiates with the user agent through to emergency assist agent. The bidding continues until the driver agent accepts one bid or rejects them all.

3.3.3 Tasks

A task is a structured set of communications and actions. The ovals denote tasks that the role must execute in order to accomplish its goal. These concurrent tasks are defined as a finite state automation specifying messages sent between roles and tasks. The lines between nodes indicate protocols between tasks, which define a series of messages between the tasks that allow them to work cooperatively.

In this step, an agent class diagram is created, as depicted in Figure 6 from the viewpoint of the roles, documented. The agent class diagram depicts agent classes as boxes, and the conversations among them as lines connecting the agent classes. Four agent classes are defined: user agent; assist agent; information agent; decision making agent.

3.3.4 Roles

- Reporting any situation that requires a police officer at the scene (e.g. assaults, traffic accident, burglary report, damage to property, parking complaint, other ordinance violations, etc.)
- Calling an ambulance for medical assistance.
- Reporting fire, smoke or fire alarm.
- Reporting a crime in progress.
- Reporting suspicious or criminal activity. (shouts for help, glass breaking, vehicle or person that does not appear to belong in neighborhood).

3.3.5 Sequence Diagrams

A use case is a narrative description of a sequence of events defining desired system behavior. A sequence diagram depicts a sequence of events between multiple roles and defines the minimum communication between roles. Use cases can be drawn from the system requirements and users. Then the use cases can be restructured into sequence diagrams. The proposed system has five main sequences of events, which are described in the following by use cases and sequence diagrams.

Refining roles: The third step is to ensure that all the necessary roles have been identified, and to develop the tasks that define role behavior and communication patterns. Through applying the use case step, the roles of the proposed system have been defined roughly, so in this step they are refined, and tasks associated with each role are created. Fig.4 illustrates a agent architecture role model.
3.5.6 Conversations

With negotiation, the agents engage in dialogue, exchanging proposals with each other, evaluating other agents’ proposals and then modifying their own proposals until all agents are satisfied with the set of proposals. Standard negotiation mechanisms adopted are based on markov chain approach and human inspired negotiations.

In negotiating phase, the roles which are emergency support agent, decision making agent, user agent includes the main messaging system. A communication diagram is a pair of finite state machines defining a conversation between two participant agent classes. The syntax of the communication class diagram is very similar to that of the roles include emergency response parameters. The assist agent realizes the negotiation as follows:

- The initiator agent estimates the minimum cost and time resource point
- The respondent agent proposes the bid lately
- The initiator agent proposes the minimum cost
- The initiator agent proposes the minimum distance

The suggested model realizes the following steps:
Step 1: Determine the asking help choices and the assist alternatives
Step 2: Revise asking resource slightly
Step 3: Diagnose whether exceeding the end time and distance
Step 4: Consider the types of messages
Step 5: Control of the limitations
Step 6: Evaluate the final situation

The agent architecture defines the configuration of the system to be implemented. The overall system architecture is defined by deployment diagrams. The proposed system is divided into two subsystems. Subsystem 1 includes four agents: agent; assist agent; information agent; decision making agent that provides the negotiable spaces of the dynamic behavior. Also some of the system components uses the some input data such as calling system people’s voice tone like calm, angry, excited, slow, rapid, soft, loud, vulgar, laughing, crying, normal, distinct, slurred, intoxicated, nasal, stutter, lisp, raspy, ragged, clearing throat, deep breathing, cracking voice, disguised, accent, electrically altered, familiar, rational, irrational; background voice frequency like, airport, animal noises, baby, clear, local, school, factory machinery, office machinery, restaurant, television, house noises, motor, music, street noises, kids, traffic, long distance, party.

3.5.7 System process

In multi agent decision making, the agents utilize classical artificial intelligence decision making methods to decision making their activities and resolve any foreseen conflicts. Emergency response criteria types are listed in Table 4.

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>Event Type</th>
<th>Event Distance</th>
<th>Event Size</th>
<th>How many people affected</th>
<th>Location</th>
</tr>
</thead>
</table>

A graph simulation has been created using Neo4j. The simulation model considers the just only fire events in Istanbul, Sisli.
Figure 5 Designed model with Neo4j

In the simulation, Istanbul (Turkey) and Sisli (a county belongs to Istanbul) has been selected. There a lot of district in Sisli but some of them used in the simulation. As shown in the image above. There are event types, streets connected to the districts, district connected, district connected to the city. Denunciations occurred at random dates are connected to the streets as the main purpose is to store them by detail address (in Figure 5). There are responses taken for denunciations, and those denunciations connected to the events. For this simulation, it is only focused on fire event.

From the simulation, system is able to interpret the inputs. System now can find fire events occurred in Sisli, responses to those events. System now might evaluate the input, look for similar response and if there is a match then it could be used as new response to that event (Figure 6).

STEP 1: Fire events from system

```
START n=node(*)
MATCH r-[rel:BELONGS TO]->n
WHERE n.name='Fire'
RETURN r
```
STEP 3: Denunciations happened in Sultan Street and taken responses

Future possibilities

- Fire events can be counted, a density map can be made from the evaluation and response plans might be developed for a specific city, county, district or street.
- System might find similar response to incoming denunciation
- A lot of scalable data can be stored
- System can learn from scalable data and generate responses
- Search and response action will be fast
- A visual map can be developed for a better understanding

Attributes can be specified and then an ordered weight function can be constructed to evaluate utility values of different solutions. The solution with the best utility will be selected. Each agent keeps a driver’s decision hierarchy as multi-attribute preferences. The people who have some problems that related about the

1. System have to determine the problem class
2. After classification, system has to find solution about the problem in a short time and short distance to someone who needs to emergency assist.
3. Emergency assist agent system has to making decision with the all of the components.

IV. PERFORMANCE ANALYSIS

The suggested system considers the different parameters for the evaluation of the information for selection of the best choice. Neo4j is an open-source database, which implemented in Java. It is able to realizes $2^{35}$ (~ 34 billion) nodes, $2^{35}$ (~ 34 billion) relationships, $2^{36}$ (~ 68 billion) properties, $2^{15}$ (~ 32 000) relationship types. Comparison of the neo4j and mysql performance with the vertex returned based on the time is given in Figure 10 and Table 11.

Size-wise Neo4j handles graphs of sizes of several billion nodes, relationships and properties out of the box. It is normal to reach read-performance of 2000 relationship traversals per millisecond (about 1-2 million traversal steps per second) fully transactional, with warm caches, per thread. With shortest-path-calculations, Neo4j is even on small graphs of a couple of 1000 of nodes 1000 times faster than MySQL, the difference increasing as the size of the graph increases.
This paper discusses agent based emergency response model with an active multi agent database system which incorporates active rules in a multi computing environment. This system helps people to reach emergency remedy resources easily. Finally, due to frequent changes in the positions and status of objects in an active mobile database environment, the issue of temporality should be considered by adapting the research results of temporal database systems area into active mobile databases.

This paper gets the agent-based simulation and determines an optimal plan to emergency response model in the shortest time possible with neo4j simulation model. Agent simulation for emergency response model improves upon other simulation models that are concerned with numerical analyses of inputs or amounts of people and structures. The agent-based system for emergency response model is grounded on empirical data taken from real-world experiments. If the agent sees an exit, it will proceed towards it and if it receives any types of direction to leave, that will be carried out without failure. Further study includes the improvement of the text mining techniques with new respect. Also agent based emergency response model provides to evaluate uncertain and vagueness information.

**REFERENCES**


Jennings, N., Gopal Ramchurn, Mair Allen-Williams, Raj Dash, Partha Dutta, Alex Rogers, Ioannis Vetsikas; “The ALADDIN Project: Agent Technology to the Rescue”.

Nick Jennings, Gopal Ramchurn, Mair Allen-Williams, Raj Dash, Partha Dutta, Alex Rogers, Ioannis Vetsikas; “The ALADDIN Project: Agent Technology to the Rescue”.

Figure 10 Comparison of the neo4j and mysql performance based on the vertex returned based on the time

<table>
<thead>
<tr>
<th>neo4j</th>
<th>time(ms)</th>
<th>vertex returned</th>
</tr>
</thead>
<tbody>
<tr>
<td>124</td>
<td>11360</td>
<td>27</td>
</tr>
<tr>
<td>922</td>
<td>162640</td>
<td>474</td>
</tr>
<tr>
<td>8851</td>
<td>2206437</td>
<td>3366</td>
</tr>
<tr>
<td>11230</td>
<td>28125623</td>
<td>49312</td>
</tr>
</tbody>
</table>

Table 5 Neo4j and mysql performance value

Kleiner, B. Steder, C. Dornhege, D. Höfer, “RoboCupRescue - Robot League Team RescueRobots Freiburg (Germany)”; RoboCup (Osaka, Japan), 2005.


