RoboCupRescue 2010 – Rescue Simulation League Team Description <S.O.S. (Iran)>

Angeh Aslanian, Aramik Markari,Faraz Falsafi
Mohammad Ehsan Soleimani Yarandi, Salim Malakouti
Navid Haeri, Reza Hoseini , Morteza Rezayi Khoshdarregi
Nima Mokhtarian, Seyed Mohammad Reza Modares Saryazdi

Robotics Research Center, Department of
Computer Engineering and Information Technology,
Amirkabir University of Technology,
No. 424, Hafez Ave.,
Tehran, Iran

angeh.a2@gmail.com,{ aramik ,faraz_cs}@aut.ac.ir
ehsan.s1987@gmail.com {salim_m}@aut.ac.ir
{navid_it, rhtl1369, m-rezayi }@aut.ac.ir
{nimamokhtarian, modaresi}@aut.ac.ir

http://ce.aut.ac.ir/~sos

Abstract. Besides working on various high level strategies of our team agents, due to the server's changes we have to work on many low level skills of our agents we think that after 7 years of S.O.S. contribution in RCRS field, we have gained enough experience which allow us to use and implement many high level AI strategies in our agents. The aim is to describe the most important and high level changes for the upcoming year so it's highly recommended to read the previous team description papers of S.O.S. in advance.

1. Introduction

The S.O.S. basic agent and its abilities and skills have been described in previous years TDPs, so this paper is to depict the new strategies added to our plan for Robocup 2010. Following will come the description of these improvements in details.

2. Agents

S.O.S. basic agents does not differ from previous years the only change is in thier implementation in order to match with the new Server, as they have the same sample abilities to apply every multi-agent strategies needed. It is suggested to review previous years TDP of S.O.S[1,2],team to find out these abilities in detail.

3. Agent skills and action selection

Agent general skills (i.e. low level abilities) basically are as the previous years, for example the path planning strategy, we use a special version of Dijkstra single source shortest path algorithm [4], using a priority queue implemented by S.O.S. team whose time complexity is $e \log(e)$ where e is the number of edges in the city graph (i.e. roads),but the implementations of our base is changed due to the simulators changes, For example Dijkstra's classic algorithm is mixed with some other heuristics in order to get the best path in the minimum time, it's good to mention that the best path is not the one that is the shortest, many other factors like unsearched buildings near by a road(the importance of this factor is described in the search section),traffic,etc ,are also involved when we try to choose the best path.

3.1 Ambulance Team Strategy in 2010

The Problem that the ambulance Team is to deal with is that, there are a number of injured civilians in the city that some of them are also buried and the ambulance agents are to rescue and carry them to the refuges. The amount of the buriedness differs for each civilian and each ambulance agent can only rescue or carry only one civilian in each cycle of simulation.

Each civilian has two other traits; the first one is its Health Point (HP).The civilian will die when the HP reaches zero. The second one is its Damage that would be subtracted from the HP in every cycle to get the new HP. Ambulances should rescue maximum numbers of civilians in minimum time.

Any team that wants to implement a good Ambulance Team agent should meet some prerequisites such as a reliable Communication system and an updated world model, an effective searching method and finally an optimized path finding method.

After examining several strategies implemented in previous years, we have done some assumptions as to be a Base of a Good Ambulance team.

- Each civilian should be rescued by only one agent, excluded some critical situations.
- Rescuing less emergency civilians is a waste of time that is avoided if there are targets that are more important.

Based on these assumptions the total strategy of our ambulance team agent would be to.

1- Collect an updated list of injured civilians that should be rescued.
2- Determine nearest reachable refuge for each injured civilian.
3- Remove the civilians that are impossible to be rescued from the mentioned list.
4- Assign priority to the remaining civilians in the list.
5- Simulate rescuing operation and change the priorities if needed.
6- Rescue civilians based on simulating result.

Besides these, two important tools are also needed in order to make this strategy effective.

- A reliable move time estimator.
- A reliable death time estimator.

Since radical changes are made in simulators we need to change are previous tools and for doing so, we are to collect samples and use neural networks instead of our previous tools.

Our strategy is centralized and if there are not any Ambulance centers, we use Fire Station as Ambulance Center and if there are no centers at all, as long as our Fire brigades have the ability to work completely distributed and independent, every Ambulance Team agent can make a greedy proper decision based on priorities.

3.2 Fire Brigade Team Strategy in 2010

We assume that the Fire Brigade's problem and its features are known to the readers so we'll discuss only the technical and important parts of the Fire Brigade's implementation.

For the upcoming year many changes have been made, that most of them are due to the simulator's changes and the remaining changes are related to fixing previous bugs and also changing some old strategies. Because the server has been changed recently and by now no strategy could be assumed to be a stable and good. The details of our strategies won't be explained and only we will have an overall look on them.

1- Strategy

As usual the fire brigade will have two strategy layers.

1- Global view. choosing a target FireZone ,and its cooperation with other agent types
2- Local view .the strategy of extinguishing the chosen FireZone effectively and the cooperation between homogeneous agents.

In past year the Fire Brigade agents didn't use Center and Messages at all, and were working Full distributed as a result in 2009 Final's last 4 maps when other teams suffered in no communication maps we maintained the same effective performance, for this year also our aim is to work with minimum amount of messages passed, but due to changes in the agents vision this approach should be changed and the agents must use some communication in order to have updated world models.
In previous year we stressed on a Perfect local strategy but again because of sever changes and mostly because of the limited vision of agents we try to be efficient in our global strategy though it's still hard to gain a good performance in local.

Global Approach

We are to add two new vital changes in our global Approach:

1. First point that we stressed on in previous year was to stick to a Fire Zone till it is completely putout and never change our target Fire Zone even if after some time we recognize another Fire Zone is more important. As experience has shown this approach is a useful strategy, but besides this strategy we’ll try to choose the best regardless of our past actions when the city is mall or the situation is critical and most of the map would be burnt.

2. Our experience shows that the number of Fires Extinguished is more important than other factors- Fire Zone position, civilians around, number of available agents , distance and etc -. So the size of the Fire Zones is the most important factor in our decision making.

Local Approach

Dividing each Fire Zone in two main parts (1st the Core – Buildings that have the most influence in fire propagation and the most energy of the Fire Zone-, 2nd the other Buildings that are not of that importance or are small enough) is one of our featured local strategies for this year.

It is obvious that, for extinguishing a fiery building heat deflects are needed. When a building is on fire and its temperature is increasing, providing this heat deflect is difficult and in many cases strong cooperation among a number of agents is needed. We aim to extinguish only new Fires and for other Fiery Buildings we cool them after they are burnt or their temperature is decreasing by itself (after they have hit their maximum point of heat) where providing the deflect is more easy and there is no need of strong cooperation between agents –cooperation among agents is costly and it wastes move and time, as the result we need more water to fill in the tanks which takes much time and so on. It goes without saying that we should avoid this vicious cycle-. It's an odd strategy but we are determined to use this new approach even at the expense of losing a bit score in some minor cases. Cooling the extinguished buildings is also a great way to control the fire and to minimize the amount of water used and also to reduce the need of cooperation among agents- Cooling is a task that can be performed by one or two agents separately -.

Our other consideration would be the fire Brigades positioning around the target Fire Zone, which could do us good through saving the time of numerous moves around the Fire Zone and helping us establishing cooperation using voice messages.

One of our most important Local Strategic changes is restricting the fires to their near most roads because statistically more than 85% of buildings can't transmit fire to the other side of road by themselves.

We would make use of other new major and minor Strategic changes (more than 100 small strategic bugs are detected), which bringing them in detail, is beyond the scope of this article.

2- Skills of Fire Brigade Agent

In real world when an experienced firefighter wants to choose a fire and extinguish it, he doesn’t make many calculations to choose a fire, he/she Just makes the Best choice he can Based on some basic features of the fires, some known facts and mostly relaying his intuition and experience, instantly; in a few words, mostly Greedy choices as previous year we aimed to implement such a thing in our fire brigades.

After implementing a working Fire agent base(based on the structure that will be discussed in Code Design part) The first step of our design is to provide our agent with some basic and accurate information, strategies and musts (How these reliable information are gathered will be discussed later) The second step is that our agent should gain some experiences, so the agents are run on several maps in order to complete their Knowledge Base(the way the Knowledge Base is completed will be discussed later). Now the agents are experienced.

When an experienced agent wants to make a decision, based on some information from the environment, it looks for a similar experience in its Knowledge Base, and repeats a good choice that it has made previously for a similar situation. This approach makes decision making very fast.

There are different testes and results of them that showed that this approach has been a simple and powerful approach for a Fire Brigade agent. Table-1 shows the result of testing different version of our fire Brigade on 2 Random Maps (these maps are chosen among maps used in Kharazmi second national Robocup competitions held in Iran).

Even though always a latter version of our code has not been better compared to earlier versions on some maps but the overall performance on a list of maps has been improved. (All documents and logs are available upon request)

Map-1			Map-2	
Code Versions	Unburned area	Time to extinguish all fires	Unburned area	Time to extinguish all fires
Before training	0.85	240	0.52	300
Trained on some Kobe	0.94	140	0.75	300
Trained on some Random	0.94	130	0.74	300
Trained on some Foligno	0.87	230	0.70	300
Trained on some VC	0.92	170	0.72	300
Trained on different maps	0.95	130	0.84	200
more different maps	0.95	120	0.86	180
Final SOS Fire Brigade	0.96	100	0.95	150

Table-1 the score of different versions of our fire brigade.

3- Gathering Basic Information of Fire

Due to limited think time (1 sec last year 0.5 -2 sec this year) we should make our choice based on some basic features and information that could be collected easily.

We Run many Empty simulations (without any agent doing something) and collect many samples from the behavior of the environment, and how in different situations it changes when we do a specific act, analyzing these samples provides many important information which is the base of your Fire Brigade's strategies.

For example the way roads surround the buildings is information that could be gained very easily but roads directly don't effect fire propagation , gathering many statistics about how buildings spread fire when they are nearby roads helped us to know what is the best way to use the relation of roads and buildings in order to extinguish Fire Zones effectively . (This was only a small part of useful information gathered, the rest could be found in our code but we don't like to publish them so easily)

Last year we gathered more than 600,000 samples from the behavior of the environment, and the same is done for the coming year (it's important to know these are not information about the parameters of the server, only they are some statistics about different features of the environment). Some analyzed information are shown in figure-2 and a small part of the sample table is shown in table-2

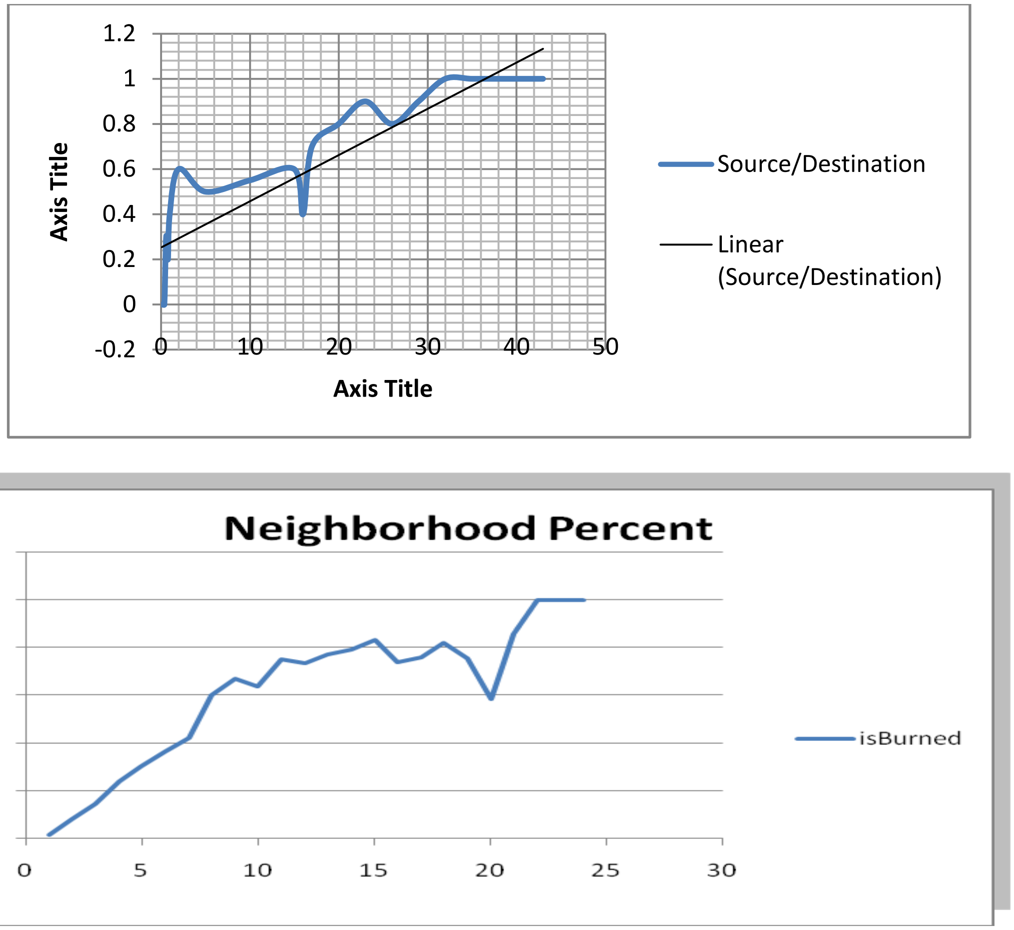


Figure -2 some figures of analyses done on samples gained from the environment

S TA	neigh	dist	Walldist	D TA	S fls	D fls	isRoad	isBurned
0.958	26	4	0	6	1	2	0	0
0.958	1	6	3	0.533	1	1	0	0
0.958	11	5	1.802	10	1	2	0	0
0.958	12	6	2.951	10	1	2	0	0
0.958	2	9	4	10	1	2	0	0
1	14	4	0.538	2	1	1	0	0
1	12	6	0.538	9	1	2	0	0
1	13	5	0.939	7	1	2	0	0
1	6	6	1.739	5	1	2	0	0

Table-2 part of the 600,000 gathered samples

4- Fire Brigades Code Design and its Knowledge Base

In previous year the most important part of The Fire Brigade was its powerful Code Design (a State-Based – Case-Based Structure).This structure made development of the code and adding new features to it extremely easy and straight forward, so that the code could be radically changed and easily developed even when few days are remained to the competition (observing the SOS_2009 s performance in preliminary round and the final round clearly shows the power of this kind of design). The structure is designed so simple, that the changes could be done even by somebody who only has some basic information about The Code.

First of all we code a working agent using this structure and Provide our agent with the simplest functions and a simple Knowledge Base Our local strategy choosing system structure is the same as the Global strategy choosing system structure is, for an example of the structure if we need to choose a building from a list of buildings the code is something like this

```
Straegychoosing system(){

        Feature1_Function (list of buildings);

        Feature2_Function (list of buildings);

        Feature3_Function (list of buildings);

        ……

        Choose building with maximum priority;

}
```

Each of these functions add some priority to buildings in the list, due to how the feature is set for each building .due to the feature we use different implementation methods such as simple if-them structure, Tables, Neural networks.

Editing this choosing system is quite easy, we can add new features, change the priorities of features (only by changing the lines … the top function has the least priority and the bottom function adds the most priority to buildings), add knowledge and experience about features in the functions.

Part of the knowledge Base is provided by neural networks which use the samples gathered from the environment (discussed previously) and also mean while the agent runs on several maps its knowledge Base is being completed.

Easily we can add and clear different features, also permute the order of lines to change their priorities and peek up the best permutation of these features. This powerful Base Design and also providing a proper Knowledge base for our agents allows the agents to make real choices and near to what a human would choose,

Due to limited pages we could not describe many of our Fire Brigades Strategic details, so for more details it is highly recommended to read the Fire team description paper of S.O.S.2009 in advance.

3.3 Police Force Team Strategy in 2010

A simple way of taking care of police force duties is to listen to reported blockades and choosing the best and most important blockade to be cleared first. But things are not to go as good as we think all the time. We may face communication-less situation. Also the enormous amount of reported blockades will complicate the decision making. Further, police force agents play a key role in searching for civilians. Therefore, we decided to design a strategy, which performs the best in all situations.

The question is how can a strategy cover all situations? Police forces are mostly listening to their channel and waiting for messages from other agents. This makes them to be one step behind. In maps without communication, police forces will not receive any reported blockades so we decided to handle these situations by guessing the other agents next task.

1- Clustering the city

At pre-compute, we partition our city into Cluster. These zones are chosen based upon the parameters below:

- There is at least one way between every junction of the zone.
- The graph of the roads and junctions of the zone is aimed to be near to a complete graph.

The idea of our clustering algorithm comes from K-means classic algorithm. Since the K-means algorithm uses geometric distances to cluster data and it is not suitable for graph of the city, and also we need to satisfy the two parameters mentioned above Therefore, instead of geometric parameters we clustered the map based on edges of the city graph..In our clustering method we choose some distinct centres by K-means and then expand them road by road (or junction by junction) .in order to have to satisfy the mentioned parameters. Then, each zone compares itself with others and tries to make itself like the others regarding the aspects of size and shape by choosing a new centroid and re-running the algorithm.[Figure-3]

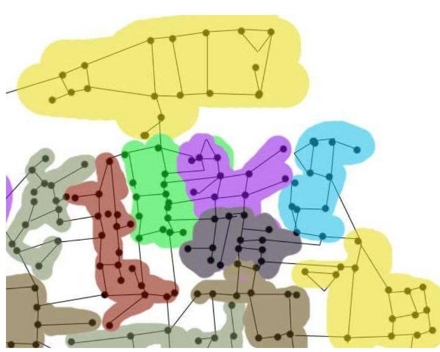


Figure-3:Dots are junctions and lines are roads.

3.3.2 Decision making process

Our Police forces' decision-making process is divided into two types: centralized and distributed systems.

"Centralized decision making system" is referred to a set of decisions, which are made in order to organize the teamwork of all Police agents for important events that need to be handled quickly. Police office assigns a task for each agent. Agents are programmed to finish these tasks before returning to their previous task. Centralized system is used while there is communication. A centralized system has two advantages:

- A center can assign agents to better task since it knows all about agents' location and previous or recent action and tasks.
- Centers receive information faster than agents thus, and are more updated.

But since this will not be useful if we run our agents on a center-less or communication-less scenarios, a distributed system is needed to be substituted.
"Distributed decision making system" refers to decisions which have to be made by agents. Therefore every Agent has a task manager which is designed to choose the task of the agents. In centralized system this task manager decides whether to take care of the task which the agent has received from its center or the task which it has in mind.
Whether centralized or distributed Agents have to decide or to be assigned to clear a zone, open the road between two zones, reachable fire zones to refuges and fire brigades, search for civilians or to search for blocked roads or etc in one zone. In one word at the first level agents choose zones as their task. When they decide which zone to work in then they will have to decide what to do first.

We used some sort of fuzzy logic in our decision making process. Every time agents want to choose a new task they estimate a value for each zone. Then the task with the highest value is chosen. These tasks are chosen depending on parameters such as the distance to the road (roads in neighbor zones have higher priority), the number and reasons of the reported blockades in a zone and the cycle in which they have been reported, size of the fire zones and etc.

This box (the algorithm) is used in both agents and centers but with different input parameters. One of advantages of choosing zones instead of requests or other kind of tasks is that are decision making process does not rely on communication system and it uses the messages just as extra information. Also it should be mentioned that these zones are not so big that they contain a lot of requests or civilians or etc.

Making the city reachable

We divided this group of tasks to three subgroups:

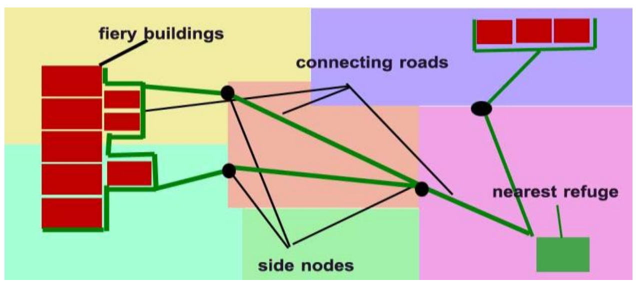
- **Connecting zones together in order to reach a global reachability**

To be independent of messages we had to find a substitute strategy for it. Before the simulations starts, we find a MST of roads in every zone. Next we find a MST through which zones are connected together. After the beginning of the simulation, agents start opening roads to connect the zones together and then opening the internal roads of each zone. This way ensure that our agents will be reachabled to all junctions of the city without being informed of the other agents' needs. But in order to optimize it we still need messages.

Another benefit that we gain by these zones is that we refuse opening unnecessary reported blockades –A reported blockade is unnecessary if there is already an open way to the destination or we can reachable the agent to its desired destination by crossing other zones or opening roads which are included in the MST. So we reduce the number of reported blockades by ignoring some of them.

- **Connecting Fire Brigades to Fire Zones and Refuge**

According to what was mentioned, we still needed messages to reach an optimize order for zones to be cleared. But we cannot risk it with fiery buildings and civilians. After a couple of cycles with help of agents' senses and their communication system we are well informed of location of fire zones. Therefore we designed and extra strategy for them. When a new fire zones has been sensed or a fire zone has been changed in size police agents take action immediately without waiting for fire agents to ask for it. We start connecting these fire zones to each other and to the nearest refuges. Also we clear the boundary roads of these building so that fire brigades will be able to move around freely and line up more effective.



Connecting roads are chosen out of the MST roads which were calculated before the beginning of the simulation. Each zone is connected to its neighbor zones using its side nodes and side roads. Side node is a node which connects to roads in different zones. Sideroad is a road which connects two junctions in different zones.

In maps with no communication since Far fires are omitted from the sense some agents search for fire zones. Also agents who are assigned to reachable the junctions keep in mind the locations of fiery buildings in their area and keep checking their condition often.

4. Search For Civilians

In order to find all Civilians buildings of the map should be searched. The buildings are partitioned in some regions. In order to choose the best region, and complete search of that region quickly small regions are used. Regions are chosen according to their priority.

Each region is evaluated by according to three factors

1- Sum of its unsearched buildings priorities.
2- Searched buildings ratio I the region

3- Other agents chosen region: in order to increase the effectiveness of our search, the aim is to assign an agent to a different and far regions form other agent's region.

Each building is evaluated by two parameters:

1- Civilian existence probability: this probability is set when some agent hears a civilian voice, as an agent hears a buried civilian's voice marks all buildings in its voice range as probable buildings that may contain that civilian and increases their priority value, or decreases their priority value when no voice is heard from its voice range.
2- Distance from fire. Buildings, which are prone to be, ignited in near future, get more priority as long as it is possible for ambulance team to save the buried civilian.

In order to maximum the effectiveness of search, another new and important strategy used is that, mean while that the agents move to their destination they try to stop in some unsearched buildings.

5. Messaging System

Due to a basic change in the centers number, different bandwidth and message failure or dropout probability, which makes arise more advantages and also difficulties to implement we have to change our Radio channel Assignment system,and other strategies.

5.1 Management

Management unit can be divided in two smaller parts:

- Central Management unit: Manages Relationship between all other parts of Message System
- Storage Management unit: Chooses message storing place

5.2 Structure

In this system, we used an XML protocol definition. All agents used their own XML file in order to define their message types[Figure-4]; it also contains the length of their tokens, and the potential senders and target agent of each type of message. Also, facilities for automatic filling of some fields and tokens with respect to the world model of the sending agent are available.

Each message box is a mixture of a hash map, a priority queue and a stack. This mixture provides our needs and also it is easy to check messages' duplication.
When a box is full we use a FIFO structure to delete excess messages.
Messages are chosen with a Greedy Strategy to be sent using a priority queue structure ,mean while the time that a message is produced has is our most important factor in choosing messages to be sent, other parameters are also involved, and a simple FIFO structure could not help do well .

5.3Acknowledgement system

Communication channels may have two type of noises added:

1. Failure noise
2. Dropout noise

So we need an acknowledgement system in order to find out which messages have been dropped. For this purpose it is necessary that messages have number. When receiver receives some messages, it saves the messages number. Then in next specified cycles, it will send an acknowledgement. Thus, sender is aware what messages are received and what messages sould be re-sent. Acknowledgement messages are similar to other messages, it has exclusive number too,in order to be re-Acknowledged

```
<SOS>
        <AgentType name="all">
                <Message name="open road">
                        <Size>90</Size>
                        <Priority>10</Priority>
                        <HasKey>True</HasKey>
                        <Data name="roadindex">13</Data>
                </Message>
                <Message name="blocked road">
                        <Size>40</Size>
                        <Priority>13</Priority>
                        <HasKey>True</HasKey>
                        <Data name="roadindex">13</Data>
                </Message>
                <Message name="Finished msg">
                        <Size>5</Size>
                        <Priority>25</Priority>
                        <HasKey>False</HasKey>
                        <Data name="search">1</Data>
                        <Data name="agentType">2</Data>
                </Message>
                <Message name="sense humanoid id">
                        <Size>45</Size>
                        <Priority>37</Priority>
                        <HasKey>True</HasKey>
                        <Data name="id">32</Data>
                        <Data name="type">8</Data>
                        <Data name="hp">4</Data>
                        <Data name="damage">10</Data>
                        <Data name="buriedness">8</Data>
                        <Data name="mlpos">13</Data>
                        <Data name="mlpos type">3</Data>
                </Message>
```

Figure-4. The XML sample file of agents

6.**References**

1. Azizpour, H. et.al:S.O.S. Team Description Paper Proceeding of Robocup 2007.
2. Ghaffuri, M et. al: S.O.S. Team Description Paper Proceeding of Robocup 2008.
3. Hashemi, B et. al: S.O.S. Team Description Paper Proceeding of Robocup 2009.
4. Cormen, T., Leiserson, C., Rivest.: Introduction to Algorithms MIT Press, Cam bridge(2000)
5. Horstmann, C: Object-Oriented Design and Patterns
6. R.C. Dubesand A.K.Jain. Algorithms for Clustering Data. Prentice Hall, 1988.