ABSTRACT
This paper describes the Intelligent Virtual Teaching Environment (IVTE) implementation. IVTE software uses the newest technologies to develop educational software. Technologies like Multi-Agent Systems, Virtual Reality, Intelligent Tutoring Systems and Distance Learning are being used in it. The paper gives emphasis in reactive agents model and implementation using Java and Java 3D API. It is going to describe an introduction in terms of Multi-Agent Systems, mainly in Reactive Agents aspects. It will be talking briefly about IVTE project, followed by Java, Java 3D API and finally, Reactive Agents in the IVTE software.

KEYWORDS: Artificial Intelligence (IA), Multi-Agent Systems (MAS), Software Agents (SA), Distance Learning (DL), Java Technologies and Applications.

1. INTRODUCTION
Nowadays Distance Learning Software in the Computer in Education category is being produced with special attention. There are scientific interests in this kind of software to improve its quality. Consequently there is a concern about how to develop efficient and cheaper educational software. Artificial Intelligence with MAS subject seems to fill this gap. According to Demazeau [1], MAS is the newest programming paradigm to develop Intelligent Agents Software. The trend in educational software is to use concepts of agents capable of feeding complex services, act and interact in the environment to reach a specific purpose. According to Russel and Norvig [2] an agent is anything that can perceive through sensors and act through effectors in the environment. Software agent can be implemented through functions to perceive and to act in the environment. In concerning to Ferber [3], an agent is a physical or virtual entity capable of acting in an environment; it can communicate directly among other agents; it is driven by a set of objectives; it has resources of its own, it is capable of perceiving its environment; it possesses skills and can offer services; it may be able to reproduce itself, its behavior tends towards satisfying its objectives. For this reason, MAS methodology is indicated to model and to implement IVTE. It is different from any other programming paradigm. IVTE models an educational software to make children aware of urban garbage correct selection. Children are aged from 8 to 10. In addition, IVTE software uses Virtual Reality techniques to simulate children’s real world.

2. IVTE ENVIRONMENT
IVTE environment [4,5] is simulated by a small village where the child makes his/her way home after a hard day at school choosing one of the possible suggested itineraries. This microworld shows characteristics that are similar to reality. The context into which the microworld is inserted represents actions that occur in the daily routine of Brazilian children.

Inside the IVTE, just like in real life, the child may follow different directions (itineraries, heuristics) from school to his/her home. All these ways give the child a chance to interact and to keep contact with the kinds of garbage from daily life and, most crucially, these ways give the child a chance to select them correctly (Figure 1).

The environment operates on a non-immersed Virtual Reality where the student has the clear sensation of being into a real environment. However, the student is only entitled to a partial view of the environment as the technology pertaining Virtual Reality allows the student to see just what is quite close to him. The environment counts on a map that allows the sense of direction positioning. By analyzing the map the student verifies his position in the game environment as well as the position of other students and the scenery elements.

The student action into the environment is adjusted by the existing elements in the scenery. The scenery is made of several ways between the school and the student residence. In these ways we can find trees, bushes, building faces, different kinds of garbage spread on the ground, peddlers. The student actions take place in two peculiar environments called outdoor and indoor environments.

All the ways that guide the student from his school to his residence are considered as outdoor environment. Along these ways or paths the student comes across different kinds of garbage that are either produced by the student himself during the interaction or come across garbage that already exists in the environment. All the garbage produced by the student in the outdoor environment is a result of his interaction with the environment such as: purchase of sweets, soft drinks from peddlers along the ways, etc.

The indoor environment comprises the student residence into which he can also interact. In this environment the student counts on rooms like kitchen and rest room (Figure 2). In the indoor environment the student can also interact with different kinds of garbage like yogurt, fruits, biscuit packages, toothpaste, paper towel, etc.

The student will also be able to count on special garbage bins or wastebaskets for the adequate selection of garbage that can be either found by chance or produced by the student himself. These special bins will be found in the indoor and in the outdoor environments. The bins used for dry recyclable items will differ in color from the bins used for wet non-recyclable items.

In this way, the child can learn about it and consequently do the correct selection. IVTE environment has many different resources to help the student learnt.

There is a device in IVTE environment which is controlled by student action. This device is presented by a tree with different developing phases, in other words, it suffers changes by time according to student action. There is another resource to give environment feedback to student, it is called ZOOM. When the student clicks in the ZOOM he/she can verify future reaction/impacts produced by his/her action in the environment. Thus, he/she verifies environment impact of his/her action 10, 20 or 100 years from now.

![Figure 1 – IVTE Main Scream](image1)

![Figure 2 – Indoor Environment](image2)
IVTE is a multi-user software, in other words, many students can act in the same time, change messages and knowledge through the environment.

Furthermore, IVTE software has an Intelligent Tutor called GUILY. He is a pedagogical agent of IVTE [6], for this reason he monitors the student action. GUILY helps students by using messages, facial and body expressions.

3. IVTE ARCHITECTURE

IVTE is based on hybrid architecture of Multi-agent System; composed by the Reactive Agents and Cognitive Agents society [7]. MAS defines agents capable to perform complex tasks. Agents can define their own objectives and action plans and perform complex interactions with other ones. In relation to this, agents paradigm has certain inherent features such as: autonomy, pro-activity, co-operation, communicability, social ability, learning, reactivity, and mobility. Multi-Agent approach can be defined as activities of autonomous agents in a multi-agent world.

An System based on Agents is similar to human society that tries to solve specific problems by itself. If it can not be able to does this, it asks for help to other members of society and thus, exchanges messages among its. According to Davidson [8] an autonomous agent is a part of a system where it is able to act independently and effectively through its sensors and effectors in the environment. Autonomous agent main aim is to perform different tasks given to it or generated by itself.

Multi-Agent System based on Reactive Agent is defined as a society of simple agents which react to stimulus, does not have memory of its past actions and thus, does not plan its future actions.

Reactive Agents are basically behaviorists. They suffer reaction after their action in the environment. They are called Stimulus/Answer Agents. They have perception and communication capacity that is proportionate by the environment. The environment is used as a way to transport messages to Agents Society.

Considering Ekdahl’s point of view [9], Reactive agents have shown their limitations. However they have shown more efficiency than Cognitive Agents to solve generic and simple tasks, and also, to implement agents features. Reactive Agents are not so versatile like Cognitive Agents, they do not have reasoning capacity that involves more than just environment perception.

4. JAVA AND JAVA 3D API REACTIVE AGENTS IN THE IVTE ENVIRONMENT

Java is an object oriented programming language, in this way IVTE basic syntax is a class with its set of classes, objects and data stored in named fields. In addition to this, IVTE has methods encoded in its structure which manipulates data. Java 3D API can be defined as a package of classes to be used with Java language. This package allows to create geometry 3D and Virtual Microworlds. According to Sowizral [10], API Java 3D is an interface of programming used in graphic applications and three-dimensional applets.

In Java 3D, graphic elements are built with objects that can be manipulated in group or individually. For this reason, objects are linked in a tree shape. This kind of structure has a description of the entire virtual universe, like geometric data, attributes and visualization to be adopted in the renderization scene. Another Java 3D API important feature is the support to Runtime Loader. It allows to Java 3D to accommodate a big amount of File format. Some IVTE implemented files are VRML files. IVTE has loader to VRML file which verifies its syntax and after that generates Java 3D object and Java code needed to interpret file content. Figure 3 shows code fragment to the loader of the object of IVTE microworld.

The IVTE reactive agents model is based on rules, that is, its actions is mapped in a set of conditions – actions rules. When one specific rule condition is satisfied then a action must happen. For example, wastebasket and garbage, in the IVTE environment, were abstracted as Reactive Agents.

```java
/*The method for the creation of 3D geometry using a form of VRML file, a Scene object to be inserted the scene, the Loader class to read the file and one string which is the name of the file in VRML */

public node ObjVrml(Scene scene, VrmlLoader loader, String string) {
    try {
        URL loadUrl = new URL(string);
        try {
            scene = loader.load(new URL(string));
        } catch (Exception e) {
            System.out.println("Exception loading URL:" + e);
            Node node = scene.getSceneGroup();
            return node;
        }
    }
}
```

Figure 3 – Code Fragment of IVTE Methods
Java constructor is a method that has the same class name and its purpose is to execute every necessary initialization of a new object.

Garbage reactive agent abstraction receives three arguments in its constructor method. One of them is the parents node needed to object implementation in the virtual microworld. Another one is a three points vector (x, y and z), which indicates its position in the virtual microworld. The last one is a type of garbage, shown in Figure 4.

```java
/** The constructor will receive the parent node to be inserted the new object; a string to identify the type of garbage; and a vector that indicating the object position in the virtual microword. */
constructorGarbage(Node_Parent node, String Garbage_type, Vector3 position)

VrmLoader;
    geometria = this.ObjVrm(scene);  
```

Figure 4 - New object constructor in IVTE microworld

Methods must be created in the class implementation to treat occurred events in the virtual microworld. Class reaction is based on a set of rules. This method shows how garbage agent receives an environmental perception based on a set of defined rules. According to environmental perception, the object state is changed. When the agent receives an environment perception some methods are activated, so the object internal state is changed through the performance of condition-action rules. The selected rules allow the methods to return to the object action. In Figure 5 shown the fragment code.

```java
// Action GarbageReactiveAgentMethod (percetion) 
/
    rule : set of rules perception-action 
    state = interpretate_input(perception)
    rule = rule_match(state, rules)
    action = rule_action(rule)
    return action
/
```

Figure 5 – Code Fragment of IVTE Methods

Consequently the agent methods are being activated. When it is activated its properties like name, type and duration time must be stored as a perception. When the child finds a wastebasket in his/her way, he/she can click over it and thus, new agents methods are activated by stored perception. The click of mouse allows to compare the kind of garbage stored to the agent condition-reaction rules.

Some agents inserted in the IVTE can generate new agents. For instance, a child does an action in the environment and this action is to eat a hotdog – *Hotdog Agent* – this generates two new types of garbage Agent. They are the *hotdog paper* and the *hotdog left*.

In the Figure 6 is exemplified some actions and reactions of IVTE Reactive Agents.

<table>
<thead>
<tr>
<th>Stimulus (reason for change)</th>
<th>Object of Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Mouse Event CLICKED</td>
<td>To buy food;</td>
</tr>
<tr>
<td></td>
<td>To store Food Agent data;</td>
</tr>
<tr>
<td>Time</td>
<td>To destroy foodAgent;</td>
</tr>
<tr>
<td></td>
<td>To create new Agents</td>
</tr>
<tr>
<td></td>
<td>To destroy Garbage Agent;</td>
</tr>
<tr>
<td></td>
<td>To use set of defined rules;</td>
</tr>
</tbody>
</table>

Figure 6 – Some Actions and Reactions of IVTE Agents

5. CONCLUSIONS AND FUTURE WORKS

The developed work is considered of fundamental importance to the IVTE project. It contributes strongly to Artificial Intelligence in Education field. Reactive Agents model brings new technologies to make the educational software more adaptive to student. New technologies implemented in it like Java allows to IVTE to develop a Distance Learning Educational software; Virtual Reality allows to IVTE to create an Educational software closer to the children’s reality; Multi-Agents systems and Software Agents allow to IVTE to give more adaptively to user and give bigger system modularity; an Intelligent Tutoring Systems allows to IVTE to promote pedagogical characteristics to software. This work helps to create
modular reactive agents. The future work is to build Authoring Software Framework directed to Educational software to implement Virtual Microworlds based on Modular Reactive agents used in the IVTE.

6. REFERENCES


