Agent-Based Stock Trader

- Paper authors:
  - Xin Feng
  - Chang-Hyun Jo

Presentation by Krzysztof Pawlak

INTRODUCTION

- This paper introduce a unique implementation scheme of the Belief-Desire-Intention (BDI) model to be used in an agent based Java application.
- The example prototype system is the Agent-based Stock Trader (AST) that is a stock-trading expert based on intelligent agents.
- Agents in AST are based on the Belief-Desire-Intention (BDI) model in artificial intelligence.
- In this work, BDI agent was build a directly in Java with the help of database that a user can handle the BDI agent dynamically by manipulating the relations among the agent's belief, desire, and intention defined in the BDI knowledge-base at runtime.

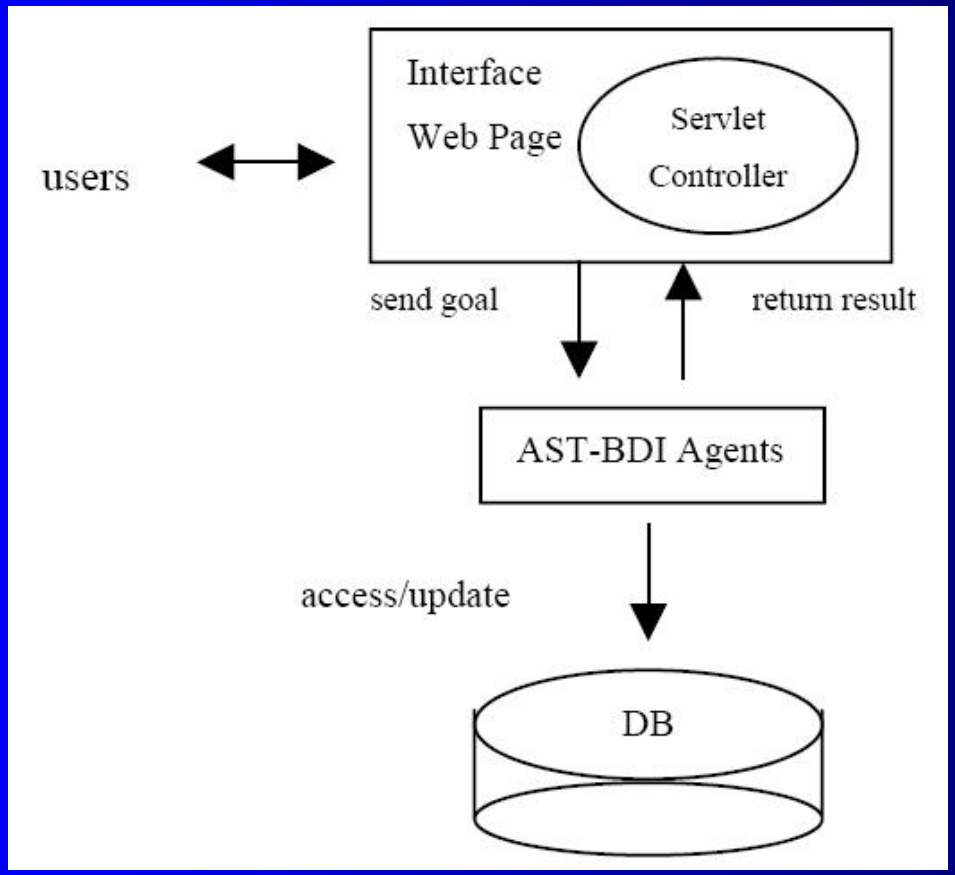
OVERVIEW OF AST

- **The BDI Concepts in AST**
- **The Architecture of AST**

The BDI Concepts in AST

- Agent-based Stock Trader (AST) is a stock-trading expert based on intelligent agents, which uses BDI model. Beliefs in AST specify all kind of stock information that agents know.
- Agents have explicit goals to achieve or events (desires) to handle. The stock names to get recommendation from expert agents in AST can be goals.
- A set of plans (Intentions) is applied to describe how agents achieve their goals based on certain beliefs. Each plan elucidates how to achieve a goal under varying environments.
- According to the technical analysis of the history and current information of some stocks, a plan will give its suggestion to help people to invest. However, another plan may be used if the history, information, or stock is different. Agent itself can autonomously decide which plan will be executed according its current situation.

The Architecture of AST



The whole system consists three parts:

•The first level is the Servlet Controller, which is a link between users and BDI agents. It provides user interface and dynamically generates corresponding web pages.

•The second level consists of the BDI agents, which are the core part of AST.

• The third level is a data access layer where all information about the AST system such as the BDI agent's beliefs and the knowledge base are stored.

•In AST, we use relational database to represent the beliefs and the knowledge base.

How it is designed

- When a user asks to find the recommendation of a specific stock, the interface agent will send this goal to the agent through the servlet program.
- Then the corresponding agent will check whether it can fulfill this goal. If so, it will choose proper plans to achieve the goal through its control structure and return the result.
- If not, it may directly ask other agents for help, or send a fail message back.
- Finally the interface agent generates the web page from which the user can get the result from the agent.

How To Implement BDI-Agents

- AST use a relational database to represent an agent's belief, including the agent's knowledge base and the environment states.
- AST has also BDI mapping table that includes current states of belief, desires to achieve, and its corresponding intentions.
- By using this BDI mapping table, there is possible manipulate dynamically belief, desire, and intention at runtime.

- The AST application starts by initiating a certain goal defined in the desire definition.
- Based on specific belief an intention is chosen to achieve the goal based on current states of belief.
- There may be several sophisticated plans for each intention. Each plan triggers the event handler to achieve the goal based on Behavioral description in each plan.
- An intention consists of a combination of one or more plans.
- To implement the concept of the BDI mapping table, authors proposed following table:

AST Database	
Table Name	**Content**
Client	Clients' personal information and account balance.
Holding	What kinds and how many stocks clients own.
Orders	All orders clients posted and the agent processed.
Knowledge	Agent's knowledge. Desire, belief values and a synthesis value of belief values.
BPmap	Shows what plan will be chosen according the certain condition.

- It is good to be able to manipulate BDIs dynamically for the BDIbased agent system.
- Belief can be dynamically changed to represent current environment. Desires can be set and changed to new ones at run time based on current situations such as belief.
- Intentions can be changed, updated, or newly added to achieve a current desire. However, it is hard to manipulate BDIs directly and dynamically well using Java.
- To solve this, the belief values can be stored in a file or database table, and can be manipulated dynamically at run time. For this AST application, belief was stored in a database table with which stock agents can consult and update dynamically.
- In the AST, the desire is also stored and manipulated dynamically in a database table. Both desire and belief are stored at the “Knowledge” table in the AST database.

- To map the corresponding plans in the intention dynamically based on both the current goal defined in the desire and the current environment based on the belief, another table “BPmap” has been used in the AST system.

- Among contents in the AST database, the table “Knowledge” is the most important table to manipulate the BDI information dynamically. The following table “Knowledge” shows an example of the knowledge base for desire and belief.

AST BDI Knowledge		
Field Name	**Data Type**	**Description**
Symbol	Text	Stock name
A	Number	World economy
B	Number	US economy
C	Number	Financial markets and institutions
E	Number	Others
F	Number	Buy or sell market
Belief	Number	The result of analysis of A~F

- The stock name “Symbol” is a kind of desire to get any recommendation on its stocks through the stock market for stock exchange (either selling or buying).
- The knowledge A through F represents its belief that represents the current environment surrounding its stock.
- The final field “Belief” holds the value that we can get from the analysis of the belief factors, A~F, by applying certain mathematical and stochastic equations.

- The following table shows the table “BPmap” that describes the mapping to the corresponding plans defined in the intention, from the information based on the current environment defined in the belief.

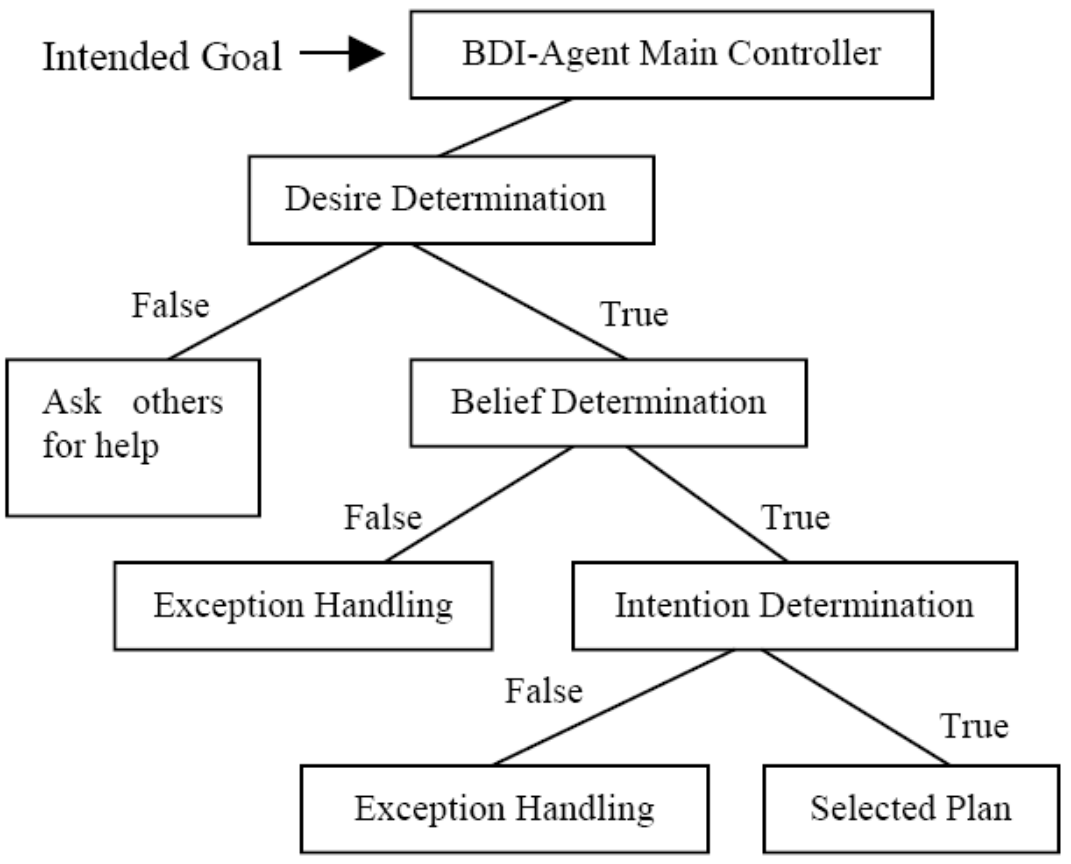
BPmap		
Field Name	**Data Type**	**Description**
LowLimit	Number	Lower limit of belief
UpLimit	Number	Upper limit of belief
Plan	Text	Corresponding plan
PlanChoice	Text	Choice of a plan

- Therefore, it is a kind of mapping table from belief to intention. The “Plan” may have recommendation values for stock exchange such as “Strongly Sell”, “Moderate Sell”, “Hold”, “Moderate Buy”, or “Strongly Buy”.

- The “PlanChoice” holds the identification number for each plan defined in the AST intention definition.

How To Program BDI-Agents

The stock export agent can be created by defining its own Belief, Desire, and Intention classes. The agent has its own main controller whose control structure is shown below.



- With the definitions of necessary belief, desire and intention using Java classes, we can create an agent class by declaring its own belief, desire, and intention from their definitions.
- When a goal is sent to an agent through the desire, the agent will check whether it can handle.
- If so, its corresponding desire will be achieved by executing proper plans defined in its intention.
- If not, it may delegate it to other agents for help.

EXAMPLE SESSIONS

- The AST system starts with the entry form of user login. The registered users may login with their user Ids and passwords. The new user may register into the system. This session is related to the table “Client” in the database.
- After successful login, a list of services regarding online stock trading that AST can provide appears.
- Currently the prototype system provides six services such as getting quotes, recommendation for stock trading, deposit/withdraw, order placement, account details, and trading history.

- The recommendation service provides a recommendation to buy, hold, or sell for a certain stock. Actually the expert agent does the back-ground work. Once a user selects a stock and clicks on the “Recommend” button, this stock name will be sent to the expert BDI-Agent as a goal through the servlet program.

- Then the expert BDI-Agent achieves the goal just as we described in the previous sections. It checks its desire and current belief to select and execute a suitable plan.

- Finally the expert BDI-Agent sends back its result, the recommendation of the stock, to the Interface through the servlet program. Both “Knowledge” and “BPmap” tables are related to this session.

- Once the agent performs a certain function needed to get an appropriate recommendation, the system shows the result. This is the corresponding answer from the export agent. For example, the AST system recommends the user to hold the stock continuously.

- User can post his stock order, and the order management agent will process it for the user. The order will be either successful or fail based on the several reasons such as the availability of the stock in the market to buy, and the availability of the funds to buy the stock, and etc. The related table in the database is “Orders”.

- The account detail option in the main menu can be selected to see the user's account in detail. It shows what stocks and how much money the user owns. The tables "Client" and "Holding" relate to this session. The user may select the trading history option to see the user's trade history in the current past. The fund manager handles both deposit and withdraw for the user.

CONCLUSIONS

- Agent-based computing is emerged as a future-computing paradigm. The BDI model is one of the powerful techniques to describe autonomous intelligent agents. In this paper authors have presented a stock trading application based on intelligent agents using the BDI model.
- This work also shows how we can implement a real world application like stock trading using the BDI-agent model to represent real world problems more naturally in a better way.