A Multi-Agent Approach for Distributed Knowledge Processing in Contact Centers

CLAUDIU IONUT POPIRLAN
University of Craiova
Faculty of Mathematics and Computer Science
A.I. Cuza 13, 200585 Craiova
ROMANIA
popirlan@inf.ucv.ro

Abstract: As contact centers grow and become more complex in their function and organisation, the knowledge processes become more formal to ensure consistency of advice and efficiency. This paper suggests a multi-agent approach for distributed knowledge processing and discusses the use of enhanced mobile agent architecture (EMA)[8] in context of contact centers to advance and frame future discussion of these knowledge intensive environments. We prove the benefits of the enterprise resource planning(ERP) management using multi-agent systems for contact centers with distributed knowledge, considering an adequate case study and providing experimental results.

Key–Words: Multi-Agent Systems, Software Agent, Mobile Agent, Contact Centers, Distributed Knowledge, Enterprise Resource Planning (ERP).

1 Introduction

The evolution of communication services over the past century has spawned a broad new industry known as electronic contact, which provides electronic communication mechanisms between people and businesses or organizations. A contact center (also referred to as a customer interaction center or e-contact center) is a central point in an enterprise from which all customer contacts are managed. The contact center typically includes one or more online call centers [3] but may include other types of customer contact as well, including e-mail newsletters, postal mail catalogs, Web site inquiries and chats, and the collection of information from customers during in-store purchasing. A contact center is generally part of an enterprise’s overall customer relationship management (CRM).

At present, most actionable knowledge about management of enterprise resource planning(ERP) systems encompasses the implementation process and it relates to software modules used in back-office or production environments. For example, in [5] is described an integrated architecture and a conceptual framework for multi-agent ERP system. However, management knowledge needs are changing as ERP adopters begin to face post-implementation performance challenges and as ERP systems extend into customer-facing or front-office work environments where the boundary between the organization and the outside world is porous and interactive.

Agent-based systems [7] claim to be next generation software capable of adapting dynamically to changing business environment and of solving a wide range of knowledge processing application. Software agents are sophisticated computer programs that act autonomously on behalf of their users to solve complex problems, and a multi-agent system is a loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each problem solver. Although sophisticated software agents can be difficult to build from scratch due to the skills and knowledge needed, the widely available agent construction toolkits may provide a quick and easy start to building software agents without much agent expertise. For example, JADE (Java Agent DEvelopment Framework) is a software Framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middleware that complies with the FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases [1]. Significant research and development into mobile agency has been conducted in recent years, and there are many mobile agent architectures available today.

In this paper is presented a multi-agent approach using EMA architecture [8] for distributed knowledge processing, in context of contact centers with enterprise resource planning(ERP) management. We introduce a multi-agent model for the analysis and process
## Contact Center Description

A contact center would typically be provided with special software that would allow contact information to be routed to appropriate people, contacts to be tracked, and data to be gathered. A contact center is considered to be an important element in multichannel marketing. Contact centers are classified by size and analyzed in relation to specific process-oriented dimensions including maturity stages. The model also points out the kinds of knowledge management (KM) interventions that are successfully established in contact centers.

The contact center architecture that includes the related components is described below and can be visualized in Figure 1. The role of the view-related components is as follows:

- **Automatic Call Distribution (ACD)** is a software feature of the telephone system that routes a call to groups of operators (also called a *queue*) based on first-in, first-answered criteria. The guiding principle is that the caller who has been waiting the longest will be first the caller routed to the next available operator.

![Figure 1: The Contact Center Structure](image)

### 2.1 Contact Center Management

A contact center offers a service, delivered through telephone calls with clients. Service level can be defined as the degree of satisfaction of callers with the offered service. This service level consists of many different aspects, related to the quality of the answer, the waiting time of the customer, etc. Some of these are hard to quantify, such as friendliness of the agent, others are more easily quantified. The main indicator for efficiency, in contact center, is the productivity [4], measured over a certain period (for example, a week). It is usually given as the percentage of time that an agent is working of his or her total scheduled working time:

$$\text{Productivity} = \frac{TWT}{TWT + TA} \times 100\%.$$  

where **TWT** is total working time and **TA** is time available.

The percentage of calls that is answered in less than a certain fixed waiting time is sometimes called the telephone service factor (TSF). Another commonly used waiting time metric is the average speed of answer (ASA). The *Erlang C* formula [4] gives the TSF, and can be used to compute the average waiting time for a given number of human agents(operators), service times and traffic intensity.

### 2.2 Typical Process in a Contact Center

The following introduces a typical process in a multi-agent call-center. Please note this process is meant to be illustrative rather than comprehensive. The customer dials the call-center number and is greeted with a number of options that include the following:

- a recorded message followed by the placement in a telephone queue managed by an Automated Call Distribution System (ACD);
- an Integrated Voice Response (IVR) that offers the caller different options where caller interacts with the IVR using a touch-tone telephone or voice control;
- the call is immediately directed by an ACD to an agent who manages the query. If the agent cannot personally resolve the query they direct the call to someone who can answer the query.

To better understand the process of a call-center that provides knowledge-based support (responses) to queries from a customer base, the following general theoretical schema, the Query-Response Cycle [10] is proffered, as shown in Fig. 2.
To manage call centers, or more generally, contact centers, effectively, one needs to have multiple skills. Roughly speaking there are those skills which are unique to the product that is delivered, and there are those skills that are needed in virtually any call center.

3 EMA Architecture for Contact Centers

In [8] the proposed Enhanced Mobile Agents (EMA) architecture improves the basic structure of mobile agents and their advantages can be highlighted very clearly in case of master/slave mobile agents system. This system includes one Master Agent and several Slave Agents [9]. The Master Agent receives complex requirements (interrogations) and create specific Slave Agents in order to process parts of a certain complex activity. In this case, the Master Agent may use Recorded Agent Information (RAI) component to adapt its behavior and/or the actions directed to Slave Agents in order to improve the solving of assignments. Also, Master Agent can use its RAI component for the same purpose and at the same time can provide access to it. We proposed EMA architecture for distributed knowledge processing in the context of contact center, as shown in Figure 3.

The proposed multi-agent ERP architecture is composed of:

![Figure 2: Query-Response Cycle](image)

![Figure 3: Using EMA in Contact Center](image)
• a set of Master Agents (a coordinating agent for each query), and
• a set of Slave Agents (data collecting agents).

Fig. 3 illustrates the abstract level of multi-agent system architecture with coordination agents serving as the representatives of each knowledge problem(query) and communicating with each other over the network. The data collecting agents (Slave Agents) perform specific tasks for a coordination agent(Slave Agent). Next, various functions/responsibilities undertaken by each type of software agent are discussed.

3.1 Master Agent - The coordination agent

The Master Agent (MA) is the heart of this multi-agent ERP architecture, is the representative of the knowledge problem when communicating with other coordination agents. The major responsibilities for the MA include:

• receiving instructions and reporting to the human operator in contact center, through an interface;
• assigning data collection to and receiving data from the Slave Agents (data collection agents);
• assigning tasks to, and receiving feedback from Slave Agents;
• communicating with and providing requested data to other MA.

With their domain knowledge, the Master Agents have the ability to monitor, communicate, and collaborate with other agents, react to various requests, as well as assign tasks to proper Slave Agents.

3.2 Slave Agent - The data collecting agent

The objective of the Slave Agent(SA) is to query specific database(s)(distributed) within the contact center servers and obtain the information requested by its own Master Agent. It possesses specific domain knowledge needed to carry out its tasks. The intelligence in the Slave Agents identifies invalid data and missing values so that the data is complete and applicable when being returned to the Master Agent. However, the structures or abilities of SA’s need not be the same in different contact centers because each contact center may have a different database management system (DBMS) or data warehouse. The responsibility of a Slave Agent is to:

• retrieve information requested by its own Master Agent;
• query distributed databases within the contact center;
• performing data analysis by running specific program and/or algorithm;
• reporting the results back to the Master Agent.

The system must be able to process the data from a distributed knowledge base. In this respect, the Slave Agents visit, one after another, all or a part of the servers to which they ask for certain information. When an SA gathers all the knowledge requested by its MA it returns home and shows the results. The system must function independently of the server addresses. Thus, in a situation where each server has a dynamic IP address, the servers list maintained by the agent must also be dynamic. The solution for this list to reflect at any moment, as exactly as possible, the addresses of the servers is to constantly update it.

4 Case Study and Experimental Results

In order to illustrate the proposed multi-agent architecture for distributed knowledge processing in a contact center, we will explain the setting and describe a scenario to answer a specific query:

A client needs to determine if the company can accept an order from customer X from city C for N units of product K at price P by Tuesday?

For simplicity, let us assume that the multinational company has four working point in four different locations: Romania (accounting department), Bulgaria (production department), Czech Republic (logistics/distribution department), and Greece (marketing department). Each working point has its own information system, database, and data architecture. The Contact Center mainframe is located in Romania and the client is located in Greece.

The solution is based on four agencies located in the considered countries (working point) - the one located in Romania is the home agency. The network topology is shown in Figure 4. Each node provide an execution environment with different hardware configurations and different operating systems. For implementation we used JADE (Java Agent DEvelopment Framework) [1].

Initially, the home agency will contain only the Master Agent. Based on the assignment it will create at least one Slave Agent to each of the other agencies. To accomplish their mission, those Slave Agents, will migrate to the appropriate agency, retrieve needed data locally, process it, and send their results. Also, each Slave Agent will record information, using RAI
component [8], about queries duration, and Master Agent will use this to improve the assignment solving. We choose a simple improvement method which will increase by 1 the number of Slave Agents sent to an agency if the recorded query duration is greater than 1 second (see Alghoritm 1).

Algorithm 1 Assignment Improve

\begin{algorithm}
\begin{algorithmic}
\For {each Slave Agent}
\If {$RecordedQueryDuration > 1$}
\State $NumberOfSlaveAgents++$
\EndIf
\EndFor
\end{algorithmic}
\end{algorithm}

By exercising domain knowledge, Master Agent organizes four tasks concurrently:

- inquire the Slave Agent in production department (Bulgaria) to obtain current inventory status of K;
- inquire the Slave Agent in logistics/distribution department (Czech Republic) for delivery information;
- inquire the Slave Agent in the marketing department for price of product K;
- monitor the status of requested information from various agents.

Due to the numbers and complexity of tasks, two Slave Agents are assigned to the production department (Bulgaria) for the product mix optimization system and for the master production scheduling system. In the logistics and distribution department (Czech Republic), Slave Agent examine a scenario of scheduling N unit of product K delivered to city C by Tuesday. In the marketing department (Greece), Slave Agent queries database and obtains current selling price (P) of product K given order quantity N and returns the result back to its Master Agent.

Upon receiving all the information from Slave Agents, Master Agent exercises its own domain knowledge to evaluate the information and to make recommendations to the client. Based on the overall results, two sets of procedures may be executed by the Master Agent:

- Case 1: All conditions are met for order acceptance
  - Master Agent informs human operator to notify the client that all conditions are met for order acceptance;
  - If the client commits the order, the operator will communicate with the Master Agent to trigger a sequence of actions;
  - If the client rejects the order, the operator will perform a sequence of actions.

- Not all conditions are met for order acceptance. In this case the Master Agent response with negative answer and can list partial solutions (for example Master Agent can evaluate the query without taking into account one of the conditions).

4.1 Experimental Results

The results presented in this subsection proves the efficiency of proposed multi-agent architecture for distributed knowledge processing in contact centers. The assignment implies to process a query that involves finding if the can accept an order that meet the following conditions:

- Customer = Renault Automobile;
- City = Paris;
- Units = 7;
- Product = summer tires;
- Price = 150 €.

In the first run the Master Agent creates only one Slave Agent for each of the Bulgaria, Czech Republic, and Greece agencies. The query duration stored in RAI component [8] of each Slave Agent are presented in Table 1.

The final result which aggregates the Slave Agents results was obtained in 1.5 seconds. When trying to understand the values of execution times we...
Table 1: Experimental results – First Run

<table>
<thead>
<tr>
<th>Czech R.</th>
<th>Bulgaria</th>
<th>Greece</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Slave Agents</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SA-C1</td>
<td>SA-B1</td>
<td>SA-G1</td>
</tr>
<tr>
<td>Duration (sec)</td>
<td>0.7</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Table 2: Experimental results – Second Run

<table>
<thead>
<tr>
<th>Czech R.</th>
<th>Bulgaria</th>
<th>Greece</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Slave Agents</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SA-C1</td>
<td>SA-B1</td>
<td>SA-B2</td>
</tr>
<tr>
<td>Duration</td>
<td>0.7</td>
<td>0.5</td>
</tr>
</tbody>
</table>

have to keep in mind that the mobile agents code is executed quasi-parallel.

The second run with the same requirements will look different, because based on information recorded in RAI component [8] of the SA-B1 agent, the Master Agent will decide to create two Slave Agents – SA-B1 and SA-B2 – which will co-operate by splitting the records in two ranges. The results of this test are presented in Table 2.

The final result on second processing of the query was obtained in 0.9 seconds.

5 Conclusion and Future Work

Intelligent business agents claim to be the next generation of model-based solutions for business-to-business and E-Commerce applications [6]. This study proposes another application for software agents as an implementation alternative of ERP, in context of a Contact Center solution. With this approach, we demonstrate that a multi-agent system can effectively improve the contact center efficiency by distributed knowledge processing, by gathering relevant information and knowledge, and more importantly, by agents cooperation in order to arrive at timely decisions in dealing with various enterprise scenarios. We provide a simple illustration to show how the proposed system might work.

Further research is needed to extend the current work, to advance and confirm this model, and to address its limitations. We intend to develop a prototype of this multi-agent system, which can demonstrate that more practical and relevant problems can be addressed successfully.

Acknowledgements: This work was supported by the strategic grant POSDRU/89/1.5/S/61968, Project IDE61968 (2009), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007–2013.

References:


