

emigo: A Large-Scale Multi-Agent Platform for the Web (Demonstration)

Rym Wenkstern, Travis Steel, Dane Kuiper,
ZW Corp
Dallas, Texas, USA
{rym, travis, dane}@myemigo.com

ABSTRACT

In this paper we present the emigo platform, an agent-based social platform for the Web. Our goal is to provide each Web user with an agent called emigo which represents the user on the Web. An emigo's role is to: a) protect the Web user from information overload; b) support and facilitate decision-making; c) Coordinate actions and decision making with multiple friends' emigos; and d) negotiate with service agents. The emigo platform functions as an ecosystem of agents (emigos, service agents, scraper agents, etc.) where each species has a specialized role and the overall operation is a result of their collective efforts. The first service offered by emigos is the "Social/Event Planner".

Categories and Subject Descriptors

I.2.11 [Distributed Artificial Intelligence]: [Multiagent systems; Intelligent agents]

Keywords

Multi-Agent Platform; Web Agents; Web Services; Collaborative and Social Computing Systems

1. INTRODUCTION

For many years, the agent paradigm has been considered as a logical choice for the implementation of personalized context-aware Web services. A wide variety of concepts, architectures and domain-specific applications for the construction of agent-based "assistants" have been developed. Most of these approaches are designed to support single users and do not require mechanisms for communication, cooperation, negotiation, etc. Given the advances in network connectivity, the development of computer applications that mediate interaction with human collectives has become a necessity and a new challenge for MAS research [2]. Even though the multi-agent research community has developed advanced models, theories and algorithms in areas such as negotiation and distributed decision making, their use has been limited to specific applications and does not extend to their engineering [2]. In addition, while a few initiatives have been launched to create global, open networks of agent

services, these efforts remained at the prototype level. Over fifteen years ago, Nwana and Ndumu [1] stated "the devil in realizing the promises of agent technology is in the details." Given the recent advances in MAS research, we state that "the devil in realizing the promises of agent technology is in the engineering." In this paper we present the *emigo platform*¹, an agent-based social platform for the Web. Our goal is to provide each Web user with an agent called emigo (i.e., electronic friend or amigo) which represents the user on the Web. An emigo facilitates decision making by filtering large amounts of data and presenting personalized results to its user. It learns its user's preferences and coordinates actions with other friends' emigos as well as negotiates with service agents to fulfill a group or an individual request. The emigo platform functions as an ecosystem of agents (emigos, service agents, scraper agents, etc.) where each species has a specialized role, and the overall operation is a result of their collective efforts (see Figure 1). The development of this platform required the application of software engineering processes, methods and techniques as well as the use of state-of-the-art development environments and tools. The platform components implement features such as distributed search, negotiation, scheduling, individual and collaborative planning, cooperative constraint satisfaction, coordination, communication and learning. The first service offered by emigos is the "Social/Event Planner" which is a Web-based software service to assist Web users (targeting young adults) with the planning, invitation, and management of activities, outings, and other activities involving one or more individuals.

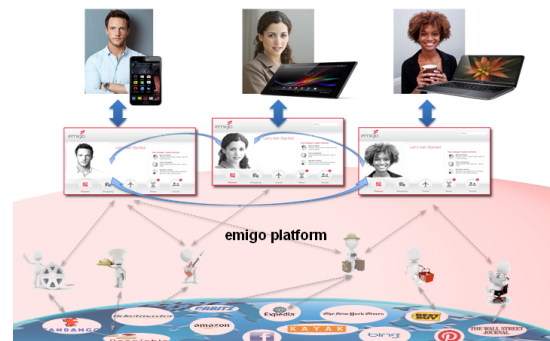


Figure 1: emigo platform

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.

Copyright © 2015, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

¹<http://youtu.be/jp46vmLLUI0>

2. ARCHITECTURE AND DEPLOYMENT

Conceptually, the emigo platform was designed as a decoupled platform that runs “on top” of the Web. The architecture includes several layers and components (See Figure 2). The software is implemented in Java and runs on Amazon Web Services cloud infrastructure.

The *Web UI* component handles user authentication and acts as a gateway for web browsers and mobile applications to interact with agent hosts.

An *Agent Host* executes and manages users’ emigos and service agents, providing agents with communication mechanisms and data interfaces. The platform is designed to support an array of hosts and scales to accommodate active user load. The *Agent Database* stores emigo information (e.g., social network, etc.) accessible solely to emigos. The *Activity Database* stores information regarding upcoming activities (e.g., restaurant, club) populated by scraper agents.

The *3rd Party Service Interface* component is a collection of libraries providing service agents with an interface for interacting with third-party services.

The *scraper agents* ensure that emigos have current information when service APIs are not available. They scrape activity information from external websites and databases, parse the gathered information and populate the Activity Database with sanitized activity information.

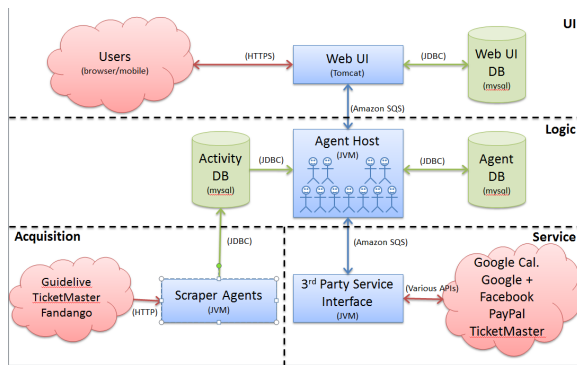


Figure 2: emigo High-Level Architecture

The current implementation of emigo leverages the scalability, availability and performance of the platform-as-a-service (PaaS) offerings provided by Amazon Web Services (AWS). The Agent Hosts run on Amazon Elastic Cloud Compute (EC2) servers to meet the current computational demand. Amazon Simple Queue Service (SQS) allows agents to interact using highly-reliable message queues. Amazon Relational Database Service (RDS) is used to store relational data used by the agents. The Multi-Availability Zone deployment and Read Replica features offered by RDS are ideal for allowing agents to collaborate efficiently with consistent information. Amazon DynamoDB offers NoSQL storage for frequently-accessed data.

3. CASE STUDY

3.1 Planning by Selecting Events

To get started, the user, Jack signs in using the email address and password provided, or through his Google+ or Facebook account. If he logged in using his Facebook or Google+ account, his personal information and friends list are imported

to his emigo account. Jacks wants to inform his emigo that he wants to see a movie between 8:30 PM and 11:00 PM. He can either select a specific movie genre, or request his agent to suggest movies that best fit his time constraint and preferences. His agent returns movie suggestions. When Jack selects a movie, his agent displays information such as the movie description, the number of amigos (i.e., friends) who have gone or are going to the movie, his agent’s explanation for the suggested event, etc. Clicking on a show time allows the event to be selected. Jack may proceed by selecting various events (e.g., dining, club) to build a plan.

3.2 Planning with your Amigos

In order to plan an outing with fourteen of his friends, Jack, the group coordinator, starts by selecting the friends that he would like to invite. Then, he selects generic event types (e.g., “movie”, “dining”, “club”) and approximate times for the outing, then asks his emigo to coordinate with his friends’ emigos to develop group plans. Jack’s emigo will interact with service agents and communicate/negotiate with each of his friend’s emigo to determine which movie, restaurant and club will have the highest level of satisfaction for the group. It presents three alternative plans. The *constrained plan* meets the group’s time constraints as closely as possible. For the *relaxed plan*, Jack’s emigo loosens the time constraints to identify a plan that the group may like better. For the *free plan*, the emigo disregards Jacks’s requests and develops a plan that it determines is better for the group (e.g., Jack may have not realized that one of the groups’ favorite singers is performing that day). The decision making for a group of fifteen people, involves over a trillion preference combinations, and is achieved by the emigo in a few seconds. If the group coordinator wishes to get a better understanding of the emigo’s thinking for the group, he can click on “Why This Event” to see the details.

4. CONCLUSION

In this paper we presented the emigo platform, a real-world, large-scale agent-based social platform for the Web. The development of the emigo platform represents a major undertaking requiring years of design, prototyping and investigation. We made use of engineering methodologies, techniques and tools to build: a) a robust and scalable platform, b) adaptable agent architectures and c) advanced communication mechanisms. So far, our efforts have focused on the development of the platform foundation with one service, but emigos have been designed to offer a variety of services (e.g., social networking, shopping) that will be added in the future. Future work includes investigating the use of more advanced machine learning and negotiation techniques. The expanding use of emigo-based services will result in an emigo becoming the electronic identity for each Web user. We hope our work will give broad visibility and appreciation to the agent paradigm and the MAS community.

REFERENCES

- [1] H. Nwana and D. T. Ndumu. A perspective on software agents research. *The Knowledge Engineering Review*, 14(2), 1999.
- [2] M. Rovatsos. Multiagent systems for social computation. In *Proceedings of AAMAS 2014*, France, 2014.