The ASK System
and the challenge of distributed knowledge discovery

Andries Stam*

Almende BV, Rotterdam, The Netherlands

Abstract. ASK is an industrial software system for connecting people to each other. The system uses intelligent matching functionality and learning mechanisms in order to find effective connections between requesters and responders in a community. Currently, Almende investigates ways to connect multiple distributed configurations of ASK to each other and to different existing systems. Thereby, we face the issue of how to derive knowledge about connections between people in such distributed heterogeneous settings. In this paper, we introduce ASK, indicate its future development and discuss the imposed challenges.

1 Introduction

ASK has been developed by Almende [1], a Dutch research company focusing on the application of self-organisation techniques in human organisations and agent-oriented software systems. The system is marketed by ASK Community Systems [2]. ASK provides mechanisms for matching users requiring information or services with potential suppliers. Based on information about earlier established contacts and feedback of users, the system learns to bring people into contact with each other in the most effective way. Typical applications for ASK are workforce planning, customer service, knowledge sharing, social care and emergency response. Customers of ASK include the European mail distribution company TNT Post, the cooperative financial services provider Rabobank and the world's largest pharmaceutical company Pfizer. The amount of people using a single ASK configuration varies from several hundreds to several thousands.

Currently, ASK configurations are deployed on a per-customer basis in a centralized manner. Developments in information and communication services, however, call for more openness and distribution of the services of ASK. Future versions of ASK likely include the possibility to connect multiple ASK configurations to each other or to different knowledge intensive systems, or even the creation of personal ASK configurations which maintain personal data and information about the connectivity with colleagues or clients. These possibilities inherently imply the distribution of data and put a challenge on the acquisition of knowledge for single or even multiple overlapping communities.

* Email: andries@almende.org

In this paper, we indicate future developments envisioned for ASK and expected challenges related to distributed knowledge discovery and management. We have set up the paper as follows. In Section 2, we provide an overview of the ASK system. In Section 3, we indicate current research initiatives at Almende and future development directions regarding ASK, and discuss the challenges regarding distributed knowledge discovery. We summarize in Section 4.

2 An Overview of the ASK System

The primary goal of the ASK system is to connect people to other people in the most effective way. The system acts as a *mediator* in establishing the contacts: people can contact the system via various media like telephone or email, and the system itself is also able to contact people via those media. In determining the *effectiveness* of contact establishment, multiple aspects play a role. For example, the rating of *human knowledge and skills* is important in cases where people request contact with specialists or service providers. In these cases, the ASK system is able to ask participants for feedback on the quality of service after the contact. This feedback can be used for optimization of subsequent requests of the same kind. A different role is played by *time schedules*, which indicate when certain people can be reached for certain purposes. The ASK system differentiates between regular plannings and ad-hoc schedules caused by sudden events or delays. Different *communication media* play another role. In most ASK configurations, voice communication (phone, VoIP) is the primary communication medium used, but different media like email and SMS are supported by ASK as well. Moreover, people can own various phone numbers and email addresses, for which they can indicate preferences and time or service dependent usage constraints. The ASK system is able to exploit knowledge about the reacheability of people via specific media, for example in the context of emergency response systems, where people must be contacted within a certain time window. In general, learning from past experiences of all kinds and forecasting based on these experiences plays a crucial role in ASK.

The software of ASK can be technically divided into three parts: the *web front-end*, the *database* and the *contact engine* (see Figure 1). The *web front-end* acts as a configuration dashboard, via which typical domain data like users, groups, phone numbers, mail addresses, interactive voice response menus, services and scheduled jobs can be created, edited and deleted. This data is stored in a *database*, one for each configuration of ASK. The feedback of users and the knowledge derived from earlier established contacts are also stored in this database. Finally, the *contact engine* consists of a quintuple of components *Reception*, *Matcher*, *Executer*, *Resource Manager* and *Scheduler*, which handle inbound and outbound communication with the system and provide the intelligent matching and scheduling functionality.

The "heartbeat" of the contact engine is the *Request loop*, indicated with thick arrows. Requests loop through the system until they are fully completed. The *Reception* component determines which steps must be taken by ASK in order

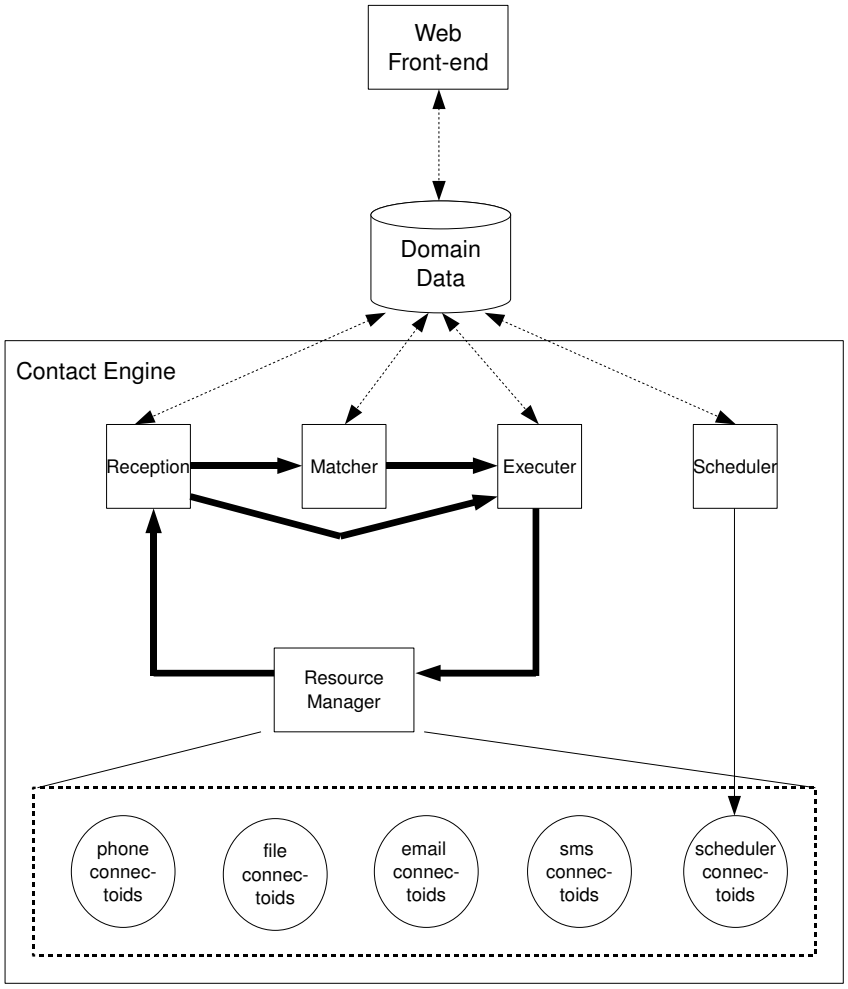

Fig. 1. ASK System Overview

to fulfil (part of) a request. The *Matcher* component searches for appropriate participants for a request. The *Executer* component determines the best way in which the participants can be connected. ASK clearly separates the medium and resource independent request loop from the level of media-specific resources needed for fulfilling the request, called *connectoids* (e.g., a connected phone line, a sound file being played, an email being written, an SMS message to be sent). The *Resource Manager* component acts as a bridge between these two levels. Finally, a separate *Scheduler* component schedules requests based on job descriptions in the database. In the next paragraphs, we discuss in more detail those components which create and exploit knowledge in ASK: the *Reception*, the *Matcher* and the *Scheduler* component.

Reception The major role of the Reception component is to determine which action should be taken by the ASK system based on a request. To give an example, if a request is received containing an incoming call event from a certain telephone number, the Reception component can decide to present a specific interactive voice response (IVR) menu to the caller, depending on the current date and time, number of the caller and the number being called. The caller is then able to provide information about the request, by selecting submenus or actions via dual-tone multi-frequency (DTMF) dial tones. A request could also originate from the scheduler, for example if the ASK system calls a user in order to ask for feedback or for availability as an ASK responder for a certain time period. The reception component is responsible for performing updates to the contents of the database in terms of adding previously unknown telephone numbers, adding feedback from users or changing schedules of responders.

Matcher The Matcher component tries to find matching users for a request. For example, a person calling the ASK system could ask for a connection with a specialist on a certain topic. Matching can be complicated, since the preferences and time schedules of the requester and candidate responders must be taken into account, as well as feedback about earlier contacts. The Matcher tries to find several candidate responders and selects between them using one of four possible methods:

1. Round Robin: the Matcher randomly selects a responder from the set of candidates available.
2. Last Spoken: the Matcher selects the responder that was selected previously.
3. Rating: the Matcher uses feedback provided by the requester about potential responders and selects the one with the highest rating.
4. Friendly Rating: the Matcher again selects based on the received ratings, but occasionally randomly selects a different responder in order to provide them with the opportunity to improve their rating.

Scheduler The Scheduler component realizes the execution of various types of scheduled jobs. Typical jobs are: contacting requesters and responders to obtain feedback about earlier connections, or contacting potential responders for availability. In executing these jobs, the Scheduler component keeps track of the time schedules and preferences of users. The Scheduler itself does not take part in the request loop: its messages enter the request loop as if they come from outside the system. Jobs for the Scheduler can be put into the database manually via the web front-end, or automatically, as the result of the execution of requests in the contact engine.

3 The Future of ASK

As we indicated in the introduction, future customers will require increasing agility with regard to the integration of new communication technologies, customer specific extensions and existing information systems in ASK. Another requirement is that of autonomic run-time scalability: the ability of the software to adapt its configuration by distribution and replication, as to achieve better load balancing without human intervention. Almende aims at even more dynamism in future configurations of ASK. Matching and scheduling functionality can be diversified and provided by multiple distributed interacting components, acting upon several distributed data sources. Matching can then be performed not only within a single community, but also across various indirectly related communities. In addition to the existing customer-specific configurations of ASK, personal contact managers can be set up to maintain personal data and knowledge about the connectivity with colleagues, clients, or service providers.

As a consequence, the current ASK system will be changed and extended considerably in the near future. Such changes can be realized in a more reliable way when combined with the use of *formal methods* for the modeling, analysis and testing of software. Currently, Almende is involved in the EU FP6 project *Credo*: Modeling and analysis of evolutionary structures for distributed services (IST-33826). In the context of this project, several formal methods and techniques are applied to the ASK system, in its current shape and with various extensions. For example, we use the modeling language Creol [5] to model the functionality of the ASK system, at a high level of abstraction, for analysis and verification purposes. Communication within and between ASK components is reorganized into communication via REO Circuits [3], as a means to achieve better compositionality through exogenous coordination and to enable dynamic reconfiguration. The mechanism of dynamic reconfiguration has been promoted by Almende in earlier research on the Common Hybrid Agent Platform (CHAP) [6]. Furthermore, we apply Task Automata [4] for the verification of timing and scheduling issues, like the completion of requests and the execution of scheduler jobs in the system.

The future scenarios we envision, and for which we are now taking the first steps, also pose high requirements on the capabilities of the ASK system to derive knowledge from inherently distributed data:

- How can efficient matching be performed in a distributed setting? Efficiency is especially important in the case of phone communication.
- How can we cope with conflicts, redundancy or dependencies in time schedules and availability data stored in different databases at different places?
- How can we exploit the updated contents from one database in the context of another database? In particular, how can we use customer feedback provided in one context to improve matching in another context?
- How can we enable cross-community knowledge discovery and at the same time keep personal contact information private and secure?

4 Summary

In this paper, we presented the ASK system, a system for effectively connecting people to each other. The system implements an intelligent matching mechanism which uses knowledge derived from past experiences. Our aim is to evolve ASK into a system in which this knowledge will be inherently distributed over various community-wide and personal databases. Formal methods play a crucial role in the modeling and verification of such an evolving system. In addition, future configurations of the ASK system certainly need to incorporate efficient techniques for distributed knowledge discovery.

References

[1] Almende website. http://www.almende.com.

[2] ASK community systems website. http://www.ask-cs.com.

[3] Farhad Arbab. Reo: a channel-based coordination model for component composition. *Mathematical. Structures in Comp. Sci.*, 14(3):329–366, 2004.

[4] Elena Fersman, Pavel Krcal, Paul Pettersson, and Wang Yi. Task automata: Schedulability, decidability and undecidability. *Inf. Comput.*, 205(8):1149–1172, 2007.

[5] Einar Broch Johnsen and Olaf Owe. An asynchronous communication model for distributed concurrent objects. *Software and Systems Modeling*, 6(1):35–58, March 2007.

[6] Jeroen Valk, Jan Peter Larsen, Peet van Tooren, and Adriaan ter Mors. Channel-based architecture for dynamically reconfigurable networks. In *BNAIC*, pages 246–253, 2005.