SPECIFICATION OF THE GOOD ENOUGH ANSWER COMPONENT

Coordinator: Pavel Shvaiko

with contributions from: Fausto Giunchiglia, Juan Pane and Paolo Besana

January 2008

Technical Report # DISI-08-005
Specification of the good enough answer component[1]

Coordinator: Pavel Shvaiko¹

with contributions from

Fausto Giunchiglia¹, Juan Pane¹, Paolo Besana²

¹ Department of Information and Communication Technology (DIT),
   University of Trento, Povo, Trento, Italy
   {pavel|fausto|pane}@dit.unitn.it

² The University of Edinburgh, Edinburgh, UK
   p.besana@ed.ac.uk

---

[1] The originally planned title of this deliverable as from the project proposal was “Specification of the semantic evaluator”. However, this new title better reflects the current contents of the deliverable and needs of the project, and therefore, is used here.
Abstract

This document provides a technical specification of the OpenKnowledge (OK) good enough answer (GEA) component. In particular, it discusses: (i) the GEA component architecture, and (ii) the GEA external interface to the other components of the OK system.

1 Introduction

The OpenKnowledge system is a peer-to-peer (P2P) network of knowledge or service providers. Each computer in the network is a peer which can offer services to other peers. OK is viewed as an infrastructure, where we only provide some core services which are shared by all the peers, while all kinds of application services are to be plugged on top of it. These plug-in applications are called the OK Components (OKCs) [1]. Notice that the OKCs link services to the OK infrastructure and may not actually contain the services themselves.

Interaction between OKCs is a very important part of the OK architecture. By using the Lightweight Coordination Calculus (LCC) [5], developers are able to define the Interaction Models (IMs) that specify the protocol that must be followed in order to offer or use a service. OKCs are the ones in charge of playing the IM roles. Different peers select to play different roles in an IM, which is then run to achieve the goals of these peers.

The purpose of the good enough answer mechanism [3] is to find good enough configurations, namely, combinations of an IM and the peers assigned to each of its roles that achieve the purposes of the peers with a reasonable investment of resources in their construction [7]. The goal of this deliverable is to provide a technical specification for the good enough answer component.

The rest of the deliverable is structured as follows. Section 2 describes an architecture of the GEA component. Section 3 presents an external interface to GEA as well as the data model for its constituent parts. Finally, Section 4 summarizes the findings of the deliverable.

2 The good enough answer component

The combination of an IM and the peers assigned to each of its roles is called a configuration. The purpose of the good enough answer mechanism is to find good enough configurations, namely, configurations that achieve the purposes of the peers with a reasonable investment of resources in their construction. It does this with the aid of two heuristic measures, namely:

• of the matching score (in the [0 1] range) between the IM roles and the peers’ capabilities [2, 4],
of some score of trust (in the \([0 \ 1]\) range) in the peers, based on their historical behavior \[3\].

The GEA component combines these two scores into a single GEA score in the \([0 \ 1]\) range. Finally, if this GEA score is higher than an empirically established threshold, then the answer is considered to be good enough, and not good enough otherwise.

Best combination of the matching and trust scores is to be obtained via empirical evaluation, including the following strategies: (i) thresholds & weighted sum of matching and trust scores, and (ii) interval-based approach, see \[3\] for details.

3 External interface

The good enough answer component offers the following interface to the other components of the OK system.

```java
public interface PeerSelectionStrategy {
    /**
     * Selects peers, using a subscription specification,
     * among a set of possible peers
     */
    public List<SubscriptionSpec> selectPeer(List<SubscriptionSpec> peersInRole);
}```
The interface supports possible selection strategies that can be implemented in the OK kernel [6]. The SubscriptionSpec class contains the specification of the subscription that a peer made in order to take part in a particular role of a given interaction model. The selectPeer method receives a List of SubscriptionSpec, that is, all the peers that want to play a particular role, and returns the best candidates to play this role.

Figure 2 shows a class diagram for the implemented strategies of peer selection (which were mentioned in §2 and described in [3]). Specifically, the GEAThresholdSelStrategyImpl class implements the thresholds & weighted sum strategy, where first all the peers with a matching score below a threshold (matchThresh), e.g., 0.5, are discarded. Then, the GEA score is computed as follows: \( T_i \times \text{weight} + (1 - \text{weight}) \times M_i \)

where \( T_i \) and \( M_i \) are the trust and the matching scores, respectively, for the \( i \)-th retained peer under consideration. \text{Weight} ranges in \([0 1]\) and is used in order to give more importance either to the matching or the trust dimension. The weight of 0.5 means that equal importance is given to both dimensions. Finally, the peer with the highest score is selected, though a top-k approach can be followed as well. GEAImplementSelStrategyImpl implements the interval-based strategy. Here peers are sorted into intervals (e.g., with a step of 0.2) or bands of width (bandWidth), according to their matching scores. Then, the best peer is the one from the highest matching band (e.g., \([0.8 1]\)) that has the best trust score which exceeds a threshold (trustThresh), e.g., 0.5.

Figure 2: The GEA class diagram.
4 Conclusions

This document has provided a technical specification for the good enough answer component of the OK system. The component is designed to be easily extensible and exploits various strategies for combining matching and trust scores. Through a first prototype we will be able to test these strategies in order to gain a better understanding of what is the best one to be used, based on the case studies within the OK testbeds, such as emergency response and bioinformatics.

References


