Expert Systems with Applications 42 (2015) 1028–1038

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Consumerized and peer-tutored service composition

Klemo Vladimir *, Ivan Budiselić, Siniša Srbljić

University of Zagreb, Faculty of Electrical Engineering and Computing, Consumer Computing Lab, Unska 3, 10000 Zagreb, Croatia

ARTICLE INFO

Article history:
Available online 28 September 2014

Keywords:
Service composition
Peer tutoring
Knowledge sharing
Consumerization of service composition tools
Expertise location
Social matching

ABSTRACT

With continued development towards the Internet of Things, services are making their way from enterprise solutions to our offices and homes. This process is a major driving force in consumerization of IT, because sustainable application development at this scale will not be possible without direct involvement and innovation from consumers themselves. In this paper, we present our work on consumerization of service composition tools. First, we describe how consumer-facing services can be presented in a usable and intuitive way. Then, combining social computing with machine intelligence, we define a recommender system that supports consumers in sharing their knowledge and creativity in peer-tutored service composition, thus empowering consumers to create their own applications. This system recommends consumers with the required service composition knowledge based on mining procedural knowledge stored in previously defined compositions. Once such a group of consumers is identified, social computing tools are used to allow them to share this knowledge with their peers. To demonstrate the effectiveness of this peer-tutored service composition model, we performed consumer satisfaction studies on our consumerized service composition tool Geppeto, which we extended with the described recommender system. Results show significant improvements in service composition in terms of performance and quality of experience.

1. Introduction

Over the last couple of years, the Web underwent a rapid process of transition from a primarily passive consumer medium to an active collaborative environment. The success of Web 2.0, an umbrella term for a series of interventions and developments in digital networking technologies we all use on a daily basis, clearly shows the way for future development. One of the most important dimensions of this new Web is the notion of commons-based peer production, as Yochai Benkler calls this paradigm shift (Benkler & Nissenbaum, 2006). In this kind of communal production, work is jointly owned and accessed by its participants who operate as peers without need for a hierarchical organization of collaboration. Since the number of Web consumers is growing rapidly and with proliferation of ubiquitous mobile devices, it is reasonable to expect that this trend of online collaboration will evolve to not only production of data and information, such as in Wikipedia, but also to production of more advanced forms of creative work like software.

The creative force of all Web consumers, called cognitive surplus by Shirky (2010), can be utilized for creation, customization and automation of software artifacts. Today, programming environments for Web consumers are mostly focused on building situational, on-the-fly applications by combining existing services. For example, web mashups are applications generated by combining content, presentation, or application functionality from disparate Web sources (Yu, Benatallah, Casati, & Daniel, 2008). This process of content combination, or remixing, is usually carried out using some kind of GUI-oriented methodology, thus circumventing a traditional textual programming interface. For instance, Yahoo!Pipes,[1] which is one of the most popular mashup editors, allows consumers to process and remix data by visually connecting various modules.

While Yahoo!Pipes focuses solely on data flow, the consumer programming tool Geppeto[2] developed in our research group extends this paradigm beyond data flow, enabling consumers to define service compositions with control flow, event flow, temporal dependencies, location awareness, communication and synchronization. In the near future, tools like Geppeto will allow consumers to create large sets of service compositions and expand community composition knowledge.

Based on current experiences in software engineering, it can be concluded that building communicating, synchronized, and

* Corresponding author.
E-mail addresses: klemo.vladimir@fer.hr (K. Vladimir), ivan.budiselic@fer.hr (I. Budiselić), sinisa.srbljic@fer.hr (S. Srbljić).

[1] http://pipes.yahoo.com
[2] http://shadowfax.zemris.fer.hr:8080/geppeto/index.html

http://dx.doi.org/10.1016/j.eswa.2014.09.033
0957-4174/

distributed service compositions that are event-, time-, and location-driven will never be easy. To address this inherent difficulty, *assistants* that help consumers in service composition, from finding and understanding component services to wiring them together, are a cornerstone of consumer computing. In this paper, we focus on such an assistant for *peer-tutored service composition*.

A prerequisite for effective peer tutoring, which aims to enable ad hoc, dynamic, and problem oriented gathering of qualified consumers, is tutor identification. A good candidate tutor should possess tacit knowledge about the services related to a problem and how they can be composed. However, people are often not fully aware of their tacit knowledge, or how this knowledge can be valuable to other consumers and how it correlates to collective knowledge of the larger community. Our research presented in this paper is based on the following hypothesis: Since procedural knowledge encoded in a service composition created by a consumer is derived from that consumer's tacit knowledge, consumers with the required tacit knowledge can be identified through analysis of the compositions they had created.

The process of consumerized service composition using a peer-tutor recommender is shown in Fig. 1. A key object used in peer-tutor identification is the *partial service composition* which a consumer created before encountering a challenge and asking for a tutor. While incomplete, this partial composition encodes parts of the consumer's tacit knowledge about the problem that is being solved. Furthermore, for many problem domains, it is likely that other consumers have already created compositions in that domain. These compositions are stored and analyzed, and are the basis for tutor identification. Specifically, the partial service composition created by the consumer asking for a tutor is compared to other compositions stored in the composition database using machine intelligence techniques. Authors of similar compositions are identified as potential tutors as they are likely to have the required tacit knowledge to help solve the problem. Once potential tutors are identified, social computing tools are used for peer tutoring with the goal of sharing service composition knowledge in the consumer community.

The rest of the paper is organized as follows. Section 2 gives an overview of related work in peer tutoring and tutor recommendation. In Section 3, we review our research in consumerization of

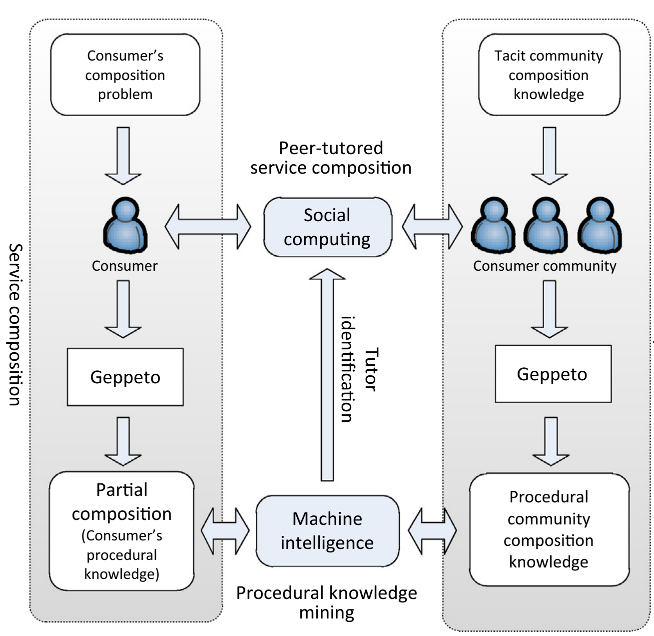

Fig. 1. Peer-tutored service composition and expertise location based on procedural composition knowledge mining.

service computing technology. We describe our consumerized service composition tool Geppeto, discuss benefits of this approach to service composition and challenges that arise in it. In Section 4, we show how peer tutoring can be integrated into a consumerized service composition environment. In Section 5, we describe how service composition procedural knowledge is encoded in the recommender, and describe the peer-tutor recommender algorithm itself. In Section 6 we briefly describe how Geppeto was augmented with the tutor recommender system. To evaluate the effectiveness of the proposed recommender, we carried out a consumer study with our students. The study setup and results are given in Section 7. Section 8 concludes the paper and proposes future research directions.

2. Related work

Our approach to peer-tutored service composition employs recent technology advancements in multiple domains: correlating different aspects of knowledge; knowledge authorship attribution; expertise location and sharing; recommendation in the context of social matching; dynamic, ad hoc, and problem-oriented social networking; and knowledge automation assistance.

2.1. Correlating different aspects of knowledge

Contemporary socially-intelligent computing, based on a synergy of computation and human intelligence, gave rise to research in knowledge management technologies that enable correlation of different aspects of knowledge: tacit and procedural, individual and collective, artificial and human, knowledge and meta-knowledge. Most of these aspects of knowledge correlation are critical for peer-tutored service composition: (I) tutors with required *tacit knowledge* about service composition are identified by *procedural knowledge* embedded in service compositions stored in developed applications, (II) *individual* tutors with required expertise are identified based on mining of *collective* procedural knowledge embedded in applications that are developed by a community of peers, (III) collaborative peer tutoring requires correlating *individual* tacit knowledge of a peer to *collective* knowledge of a community of tutors, and (IV) the *artificial meta-knowledge* derived from procedural knowledge mining is correlated to *peers' (human) tacit knowledge* to enable correct tutor identification. Although knowledge representation, correlation, management and transfer have been in research focus for a long period of time (Liao, 2003), recent technology development enables pragmatic use of this research in various areas such as in search engines (Bobick & Wimmer, 2012), the semantic web (Berners-Lee, Hendler, & Lassila, 2001), and human computation (Quinn & Bederson, 2011).

2.2. Knowledge authorship attribution

Source code authorship attribution is an example of author identification based on procedural knowledge. Authorship attribution is defined as the process of assigning authorship of an unattributed or contentious sample of work to its correct author amongst a finite pool of authors (Burrows, 2010). In his thesis, Burrows introduces authorship attribution as a subset of fields such as software forensics and plagiarism detection. While knowledge authorship attribution is usually used to differentiate between authors' coding styles, we research these methods for the purpose of service composition matching.

2.3. Expertise location and sharing

Several knowledge-based recommender systems are described in the literature that depend on the explicit domain-specific