# Collaborative Adaptive Case Management with Linked Data

Sebastian Heil
Technische Universität Chemnitz
Chemnitz, Germany
sebastian.heil@informatik.tu-chemnitz.de

Stefan Wild
Technische Universität Chemnitz
Chemnitz, Germany
stefan.wild@informatik.tu-chemnitz.de

Martin Gaedke
Technische Universität Chemnitz
Chemnitz, Germany
martin.gaedke@informatik.tu-chemnitz.de

## ABSTRACT

An increasing share of today's work is knowledge work. Adaptive Case Management (ACM) assists knowledge workers in handling this collaborative, emergent and unpredictable type of work. Finding suitable workers for specific functions still relies on manual assessment and assignment by persons in charge, which does not scale well. In this paper we discuss a tool for ACM to facilitate this expert finding leveraging existing Web technology. We propose a method to automatically recommend a set of eligible workers utilizing linked data, enriched user profile data from distributed social networks and information gathered from case descriptions. This semantic recommendation method detects similarities between case requirements and worker profiles. The algorithm traverses distributed social graphs to retrieve a ranked list of suitable contributors to a case according to adaptable metrics. For this purpose, we introduce a vocabulary to specify case requirements and a vocabulary to describe skill sets and personal attributes of workers. The semantic recommendation method is demonstrated by a prototypical implementation using a WebID-based distributed social network.

## Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

## General Terms

Algorithms, Languages, Management

## Keywords

ACM, Linked Data, Social Web, Expert Finding, WebID

## 1. INTRODUCTION

Driven by the information age an ever increasing share of today's work is considered as knowledge work. The nature of this type of work is not only collaborative, emergent, unpredictable and goal-oriented, but also relies on knowledge and experience [7]. Traditional process-oriented Business Process Management (BPM) is not well applicable to areas with a high degree of knowledge work [9]. Addressing this issue, non-workflow approaches [1], in particular Adaptive Case Management (ACM), increasingly gain relevance [3].

ACM systems assist knowledge workers by providing infrastructure to handle dynamic processes in a goal-oriented way. While traditional BPM solutions plan processes in advance, ACM systems enable adaptivity to unpredictable conditions. Adaptivity is accompslihed by allowing for planning during execution. Cases represent instances of unpredictable processes and aggregate all relevant data. For adapting a case to emergent processes, ad-hoc goals can be added. Not necessarily all of them can be achieved by persons currently involved. Further workers may be required to contribute.

Finding suitable workers, however, still relies on individuals. They have to perform a selection depending on the requirements at hand. This requires knowledge of potential contributors and their experience. Given that the complexity of selection increases with the amount of work requirements and eligible contributors, manual assignment does not scale well, especially not with web-scale processes [5]. As a consequence, work is often assigned to workers who are not the most suitable among all available. This can cause increased times for completion and outcomes of decreased quality.

Automated support for finding and addressing knowledge workers to contribute to a case is required. If a part of the work (i.e., a goal of the case) cannot be accomplished, it is necessary to identify suitable knowledge workers based on the skills and experience required for that particular part.

In this paper we demonstrate CRAWL, an approach for Collaborative Adaptive Case Management with Linked Data. It leverages Web technology, WebID and RDF in particular, to automatically identify experts to contribute to an ACM case. To achieve this, our three main contributions are:

1. A vocabulary to add existing/required knowledge worker experience to WebID profiles/ACM cases.
2. An algorithm utilizing Linked Data to find suitable workers based on their experience and case requirements.
3. Demonstration of CRAWL by integration into a WebID identity provider and an ACM system.

As knowledge work becomes an increasingly important and widespread part of work [4] and ACM evolves as an approach addressing this type of work, we are convinced that enabling knowledge workers to find the right collaborators to con-


*WWW'14 Companion,* April 7–11, 2014, Seoul, Korea.
ACM 978-1-4503-2745-9/14/04.
http://dx.doi.org/10.1145/2567948.2577030.

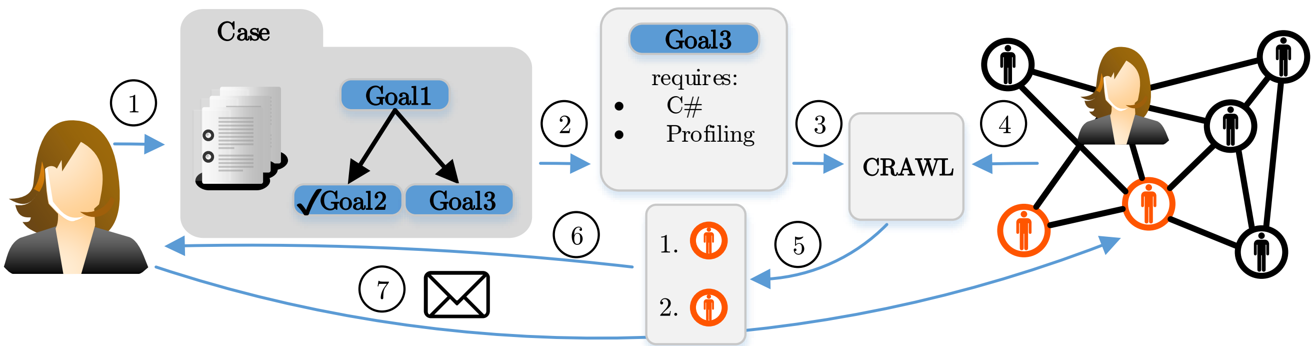


**Figure 1: CRAWL Overview**

tribute to multi-disciplinary cases impacts the performance of future enterprises [1].

The paper is organized as follows: We showcase the application of CRAWL in a usage scenario in Section 2. We present details of the approach in 3. We position our solution to related work in Section 4 and conclude the paper in 5.

## 2. EXPERT FINDING WITH CRAWL

In this section we describe how CRAWL assists expert finding in ACM systems. The scenario shown in Figure 1 demonstrates our approach. Casey is a second-level-support worker employed with a software development company. A key customer has reported a bug in a software product that is developed by the company. Casey is responsible for handling this support case. She uses an ACM system to assist her work. As she investigates into the problem, she sets diverse goals and asks experts from third-level-support to contribute. At some point a detailed profiling is required to check for concurrency issues. However, there is no expert on this topic available. To assist Casey in finding such expert, CRAWL facilitates the following workflow (cf. numbers in Figure 1):

1. Casey adds a corresponding goal to the case.
2. Casey defines requirements (e.g., C# and Profiling).
3. Casey starts CRAWL.
4. CRAWL traverses Casey's social graph.
5. CRAWL generates a list of eligible workers.
6. Casey selects the most suitable candidates.
7. Casey asks them for contribution to the goal.

A distributed system implementing CRAWL might provide user interfaces similar to Figures 2 and 3. In 2, Casey adds skills to her profile using WebID identity provider and management platform Sociddea [11]. In 3 the VSRCM[1] case management system provides her with a list of recommended candidates with their skills and contact information.

Further information on our solution including a screencast is available at:
`http://vsr.informatik.tu-chemnitz.de/demo/crawl`

## 3. THE CRAWL APPROACH

This section provides details of steps 4 and 5 from Section 2. Figure 4 shows the traversal, rating and candidate recommendation. The required skills $r_0, r_1, r_2$ of Goal3 and Casey's social graph are the input in accordance with the scenario from Figure 1. In this example, Casey knows $B$ and $C$. $B$ and $C$ know $D$, $C$ knows $E$. CRAWL has already rated $B$ with $R(B) = 15$, $C$ with $R(C) = 0$ and $D$ with $R(D) = 10$. To get the rating of $E$, the similarities between required skills and existing skills are calculated using linked open data. The WebID profiles with their ratings are stored in a triple store. Using SPARQL, the ordered list of candidates is generated.

Linked Data provides CRAWL with a large knowledge base for concepts describing skills. CRAWL references this data to describe a) existing experience for persons and b) experience required to achieve a case goal or contribute to it. To store the skill references for a) we employ WebID profiles. WebID profiles are essential artifacts of the WebID identification approach. They contain an identity owner's personal data described in a machine-readable way using Linked Data. With WebID, users are enabled to globally authenticate


**Figure 2: Skill definition in Sociddea**

[1]VSR Case Management, cf. `http://vsr.informatik.tu-chemnitz.de/demo/vsrcm/`

themselves, connect to each other, manage their profile data at a self-defined place and specify customized views [10]. In order to accommodate the data for a) within a WebID profile we introduce the following vocabulary: The RDF property `vsrcm:experiencedIn` connects a `foaf:Person` with a URI which represents this person's experience in *something*. The data for b) is stored in the case descriptions, connecting a goal via `vsrcm:requiresExperienceIn` with a URI. For referring to the actual skills for both a) and b), we propose the URIs to reference concepts which are available as dbpedia[2] resources. With dbpedia being a central element of the LOD cloud[3], this intends to increase the degree of reusability and extensibility of data we add to cases & profiles.

Supporting users in specifying their expertise and case requirements, we extended the user interfaces of Sociddea (cf. Figure 2) and VSRCM to allow specifying skills using regular English words. We use prefix search of dbpedia lookup service[4] to match user input against dbpedia resources. A list of skills is updated live as the user is typing. The resulting user interaction is known from platforms such as LinkedIn[5].

Finding suitable workers requires a traversal of the requestor's social graph established by `foaf:knows` connections. The traversal algorithm is implemented as a depth-limited breadth-first search. It dequeues a WebID URI identifying a person, retrieves the corresponding WebID profile, calculates the rating $R$, marks the WebID URI as visited and adds all unvisited WebID URIs referenced via `foaf:knows` and their depth value to the queue. The initial queue consists of the WebID URIs of the persons already involved in the case, their depth value is 0. As recent studies indicate an exponentially rising number of nodes in a social graph with increasing depth, maximum depth is introduced. Additional limits can be the number of WebID URIs already visited or the number of suitable candidates with rating above a certain threshold. The WebID-based distributed social network is relatively new and small compared to Facebook, Google Plus, Twitter etc. Currently the limit is set to 5.

[2] `http://dbpedia.org/`
[3] `http://lod-cloud.net/`
[4] `http://wiki.dbpedia.org/lookup/`
[5] `https://www.linkedin.com/`

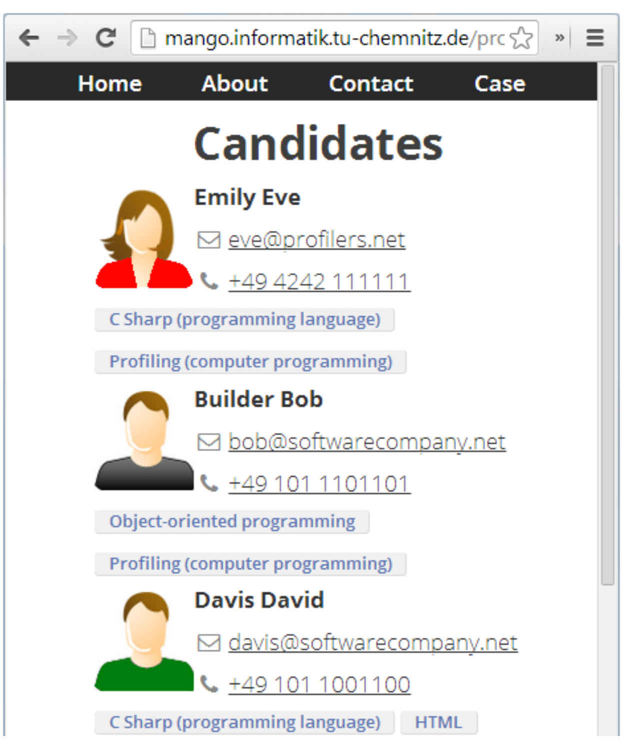


**Figure 3: Candidate recommendation in VSRCM**

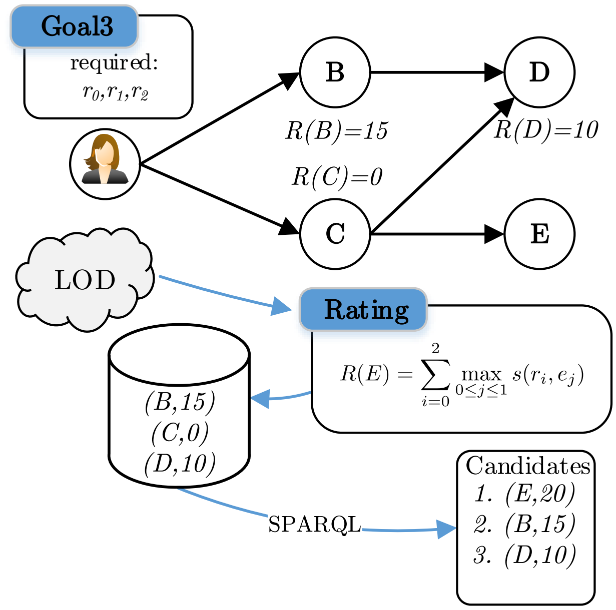


**Figure 4: Traversal, rating and recommendation.**

For the proof-of-concept demonstration, we use a prototypical rating function adapted from [6]. The set of concepts stating the required skills $\{r_1, \ldots, r_m\}$ is compared to the set of concepts describing the existing skills of each candidate $E_C = \{e_1, \ldots, e_n\}$. Both sets are represented by sets of dbpedia URIs. The similarity $s(r, e)$ between two concepts is calculated distinguishing different types of concept matches:

1. Exact Concept Match - URIs are identical: $e = r$
2. Same Concept As Match - URIs are connected via `owl:sameAs`: $r$ `owl:sameAs` $e$
3. Related Concept Match - URIs connected via `dbprop:paradigm`, `dcterms:subject`, `skos:narrower` etc.

These concept match types can easily be extended to facilitate an adapted rating. The basic idea is that each type yields a different similarity rating. For the sake of this demonstration, we use the following values: 10 for 1), 9 for 2) and 5 for 3), otherwise 0. Finding more precise values requires further empirical evaluation. For each pair $(r_i, e_j)$ the similarity $s(r_i, e_j)$ is computed. To calculate the candidate rating $R$, only the maximum similarity per required skill is considered:

$$R(C) = \sum_{i=0}^{m} \max_{0 \leq j \leq n} s(r_i, e_j) \quad e_j \in E_C$$

The WebID profile graphs of all visited candidates are added to a triplestore. For each of them, a statement containing the calculated rating is asserted into the graph. The final ordered list of rated candidates results from executing the SPARQL query shown in Listing 1 on the triplestore.

```
SELECT ?candidate ?rating
WHERE { ?candidate a foaf:Person .
  ?candidate vsrcm:rating ?rating .
  FILTER( ?rating > ?minRating )}
ORDER BY DESC( ?rating )
```

**Listing 1: SPARQL query for candidates.**

The first implementation of this rating algorithm showed performance issues. Evidently, sequential traversal of WebID profiles and rating calculation have a huge impact on performance due to the high number of HTTP requests they trigger. To improve this, we adapted the algorithm to concurrently retrieve and rate the WebID profiles. Another improvement is caching for the dbpedia resources describing skills and for the results of pairwise concept similarity comparisons between required and available skills. These adjustments could improve performance by factor 2.

Having retrieved and rated a subset of the social graph, CRAWL presents a list of recommended candidates and contact information to the person initiating the search (cf. Figure 3).This step allows for later extension to enable constraint criteria to be applied, for instance, to filter candidates from a specific company, within the same country etc.

# 4. RELATED WORK

Our approach is an application of the *social routing principle* [5] to the ACM domain. Unlike the idea of task delegation through an open call known from Crowdsourcing research [2], we follow the idea of inviting suitable experts to contribute to a case by utilizing social graphs. The *conceptual routing table* described by Dustdar et al. is formed by `foaf:knows` statements and contact information in WebID profiles.

The crowdsourcing scenario described by Schall in [8] is similar to our approach in that work items are outsourced to handle them by suitable experts. This Process Flow / Crowd Flow (PFL/CFL) scenario follows a task-oriented organization with associated open calls for contribution, whereas CRAWL enables to find skilled experts for accomplishing goals. Contrary to our approach, PFL/CFL does not use Linked Data for describing requirements and experience, for discovering eligible workers, and for matching their expertise.

Web-based task management applications like Trello[6] and Wunderlist[7] allow defining sets of work items, assigning them to workers. For organizing and assigning tasks, responsible persons rely on prior knowledge about workers, i.e., there is no support for automatically incorporating individual social graphs or considering a worker's capabilities, as done in CRAWL. Compared with goal-oriented ACM systems, their rigid organization using non-connected tasks is rather inflexible when it comes to incorporating conditional changes.

# 5. CONCLUSIONS AND FUTURE WORK

In this work, we presented the CRAWL approach leveraging web technology for finding eligible workers to contribute to ACM cases. It comprises a vocabulary for skill definition in WebID profiles referencing dbpedia resources, a method for traversing distributed social networks based on foaf:knows relationships and an extensible rating function for WebID profiles. We demonstrated CRAWL by implementation based on the WebID identity provider and management platform Sociddea and the case management system VSRCM.

Our future research interest will be to express and rate skill endorsements. If an experience statement in a candidate profile has been endorsed by someone else, this should have higher impact on the rating than un-endorsed statemets. Also, rating can be sharpened by adding new concept match types. Empirical data will help to adjust the similarity rating values and define limit for the traversal. Machine learning can be used to provide adapted parameters. Moreover, further work is needed to inquire the possibility to convert the algorithm into a MapReduce variant which would allow to run in the Hadoop environments of the major cloud providers.

[6] `http://www.trello.com/`

[7] `http://www.wunderlist.com/`

# 6. REFERENCES


[1] I. Bider, P. Johannesson, and E. Perjons. Do workflow-based systems satisfy the demands of the agile enterprise of the future? In *Business Process Management Workshops*, volume 132 of *Lecture Notes in Business Information Processing*, pages 59–64. Springer Berlin Heidelberg, 2013.

[2] D. C. Brabham. Crowdsourcing as a Model for Problem Solving : An Introduction and Cases. *Convergence: The International Journal of Research into New Media Technologies*, 14(1):75–90, 2008.

[3] C. L. Clair and D. Miers. The Forrester Wave$^{TM}$: Dynamic Case Management, Q1 2011. Technical report, Forrester Research, 2011.

[4] T. H. Davenport. Rethinking knowledge work: A strategic approach. *McKinsey Quarterly*, 2011.

[5] S. Dustdar and M. Gaedke. The social routing principle. *Internet Computing, IEEE*, 15(4):80–83, 2011.

[6] H. Lv and B. Zhu. Skill ontology-based semantic model and its matching algorithm. In *2006 7th International Conference on Computer-Aided Industrial Design and Conceptual Design*, pages 1–4. IEEE, 2006.

[7] N. Mundbrod, J. Kolb, and M. Reichert. Towards a System Support of Collaborative Knowledge Work. In *Business Process ManagementWorkshops: BPM 2012 InternationalWorkshops*, pages 31–42, Tallin, Estonia, 2012. Springer Berlin Heidelberg.

[8] D. Schall. A human-centric runtime framework for mixed service-oriented systems. *Distributed and Parallel Databases*, 29(5-6):333–360, 2011.

[9] K. D. Swenson. Position : BPMN Is Incompatible with ACM. In *Business Process ManagementWorkshops: BPM 2012 InternationalWorkshops*, pages 55–58, Tallin, Estonia, 2012. Springer Berlin Heidelberg.

[10] S. Wild, O. Chudnovskyy, S. Heil, and M. Gaedke. Customized Views on Profiles in WebID-Based Distributed Social Networks. In *Web Engineering*, volume 7977 of *Lecture Notes in Computer Science*, pages 498–501. Springer, 2013.

[11] S. Wild, O. Chudnovskyy, S. Heil, and M. Gaedke. Protecting User Profile Data in WebID-Based Social Networks Through Fine-Grained Filtering. In *Current Trends in Web Engineering*, volume 8295 of *LNCS*, pages 269–280. Springer International Publishing, 2013.