

A Similarity Measure for Task Contexts

Roza Shkundina, Sven Schwarz

R.Shkundina@mail.ru, Sven.Schwarz@dfki.de
German Research Center for Artificial Intelligence (DFKI GmbH)
D-67608 Kaiserslautern, Germany

Abstract. Knowledge workers are often embedded in an organizational setting, where execution of processes allows for appropriate, context-sensitive support. When a knowledge worker starts a new task in a workflow management system, chances are good, that someone of his colleagues has already performed or is currently busy with a similar task. Our research aims at enabling a knowledge worker to make use of workflow tasks in a minimally disturbing way. Therefore, we have to elicit his task context and find out workflow tasks with similar context. As such a task context consists of a set of contextual elements, a technique to compare two task contexts and the contextual elements therein is needed. This paper presents a similarity measure for these contextual elements, and hence, for task contexts.

1 Introduction

Knowledge workers are often embedded in an organizational setting, where they execute a multitude of implicit, or even better, explicit processes. The workflow management paradigm allows for explicitly modeling, processing, controlling and archiving the explicit processes. Furthermore, it can be used to build support systems for knowledge workers by providing rich, context-sensitive, pro-active support during their work [3,4].

Short side remarks: We distinguish the term “process” (the real-life process, the user executes) from the term “workflow”, which is a machine-readable and processible representative of the real-life process. In the FRODO project [5] we created and realized the Weakly Structured Workflow paradigm [4], which for instance allows for a hierarchical decomposition of workflow tasks. We did not keep with using the term “workflow activity”, instead, we used “workflow task” as a representative either for an atomic workflow activity or for a hierarchy of subtasks. That way the term “workflow” only designates a special “workflow task”, namely a root task.

One research topic in the EPOS¹ project aims at eliciting a knowledge worker’s context by observing her behavior (operations in applications and OS) [1]. This user context will then be used to estimate potentially relevant workflow tasks. That way,

¹ This work has been supported by a grant from The Federal Ministry of Education, Science, Research, and Technology (FKZ ITW-01 IW C01) – see <http://www.dfki.de/epos/>

potential task switches can be recognized and the WfMS can be triggered (semi)automatically with minimally disturbing user interaction. Apart from a smart, light-weight interface to the WfMS, this also allows for suggesting and presenting the knowledge workers related workflow tasks and then providing them with corresponding, pro-active information or recommendations. These recommendations will aid the user taking more effective solutions for her knowledge-processing task. The estimated related tasks can be both running and finished ones, and they can also be tasks from colleagues. So, we are not only aiming towards a smart task switch recognizer, but also towards informing a knowledge worker about related tasks and knowledge of her colleagues, as well as, reusing old, finished workflow tasks. That way, individual process know-how (which has been conserved) gets reused pro-actively. On the other hand, new useful resources utilized for the execution of some task can be stored immediately in the workflow management system for later reuse.

The most important question and the topic of this paper is, of course, which task is *relevant* for a user's current context. The question of how to elicit the user's context (by user observation) is already tackled by [1]. [7] describes how to model a user's context in a (personal) knowledge management scenario, and [6] shows how workflow context is acquired and modeled to enable context-sensitive, pro-active support. Now, this paper fills a gap by presenting an approach to calculate the relevancy of workflow tasks for the user's current context.

This paper is structured as follows: Section 2 makes some remarks about the data structures present and used for modeling "context" in our scenario. The term "task context" is defined herein. Section 3 goes into detail regarding the types and attributes of contextual elements. This section is one of the main parts of this paper. The other main part is section 4, which derives a similarity measure for the contextual elements, and hence, for two tasks contexts. Section 5 gives a short information about ongoing implementation efforts. Section 6 gives an example of the similarity measurement for task contexts. The paper concludes with section 7.

2 Data Structures For Context

When talking about *relevancy* of workflow tasks, we have to look at them first. There is lots of information available concerning a workflow task. There are, for example, the task descriptions (an informal text) and the task structure (hierarchy). The tasks also provide related information (documents, notes, links to web-pages). The performer of a task conveys interesting information, too. The knowledge workers are assigned to the tasks either via direct delegation or due to a specific role or skill requested to do the task.

[6] used aspect-oriented modeling to capture the "workflow context" enabling workflow-specific, context-sensitive support. Following the same methodology EPOS analogously uses aspect-oriented modeling to capture the "user context" enabling personalized, context-sensitive support. Thus, the data structures are already similar. However, in order to calculate the relevancy of a workflow task for some user context, we extract a *compatible* view out of both: We define **task context** to keep (a) the user-relevant parts of the workflow context, as well as, (b) the workflow-relevant

parts of the user context. Using this shared context model we can reduce the relevancy calculation to a similarity measure. We have to solve the following problems:

- extraction and integration of workflow context into task context;
- extraction and integration of user context into task context;
- similarity measure/calculation for task contexts.

As workflow context and user context have the same data structure, we handle the task context view and its contextual elements in the following section.

3 Contextual Elements

At present, the following contextual aspects are interesting for the task context:

- informational aspect: relevant and touched documents, accessed or created by the user during a task; subject or corresponding domains/topics;
- organizational aspect: organizational structures involved into or relevant to the task: person(s), roles, skills, interests of a user, projects and organizational units she belongs to;
- behavioral aspect: the behavior of the user – her performed operations and actions;
- operational aspect: applications and tools used by the user (to accomplish her task);
- causal aspect: task goals and estimated user goals;
- chronological aspect: timeline of events occurred in the system, e.g., recently processed workflow tasks.

The information about real world objects, people, documents or events is represented in each of the aspects by instances of corresponding resources or concepts, defined in turn in context ontology. Such informational entities are wrapped by so called contextual elements. Besides the wrapped entity a contextual element also contains a value for the confidence and an explanation about its existence.

General Remarks on the Context Ontology

Contextual elements are not loosely, unrelatedly packed into the context. Moreover, semantically rich relations between the contextual elements are modeled. These relationships are conveyed via the context ontology.

As an example, *E-Mail*, *User*, and *Project* are all parts of the ontology, potentially having relations between them. A *User* may have written some email. Additionally, that *User* may be a member, or even the leader, of some project. Another example: An *E-Mail* may be about some *Project* (either this is classified manually or the subject or email body gives good evidence). Another important thing is the usage of organizational or group-wide domain ontologies. A taxonomy of *DomainConcepts* (*D.C.*) resembles a domain ontology, providing two relations, *is-a* and *part-of* for example.

Contextual elements have slots/attributes related to them holding further information, e.g., a *User* has hat attributes describing her skills and interests.

1. The informational aspect consists of elements that contain and classify information, related to the current task, i.e., information mediums – files, folders, notes etc. – and domain concepts derived from them. The latter helps to consider

the linguistic counterpart of similarity. To derive them from the textual data, we perform linguistic and structural analysis. Among the possible sources of information are: a) document metadata: meta-tags introduced into a HTML document, metadata of an Office document, headers of an E-Mail message; b) document textual body: Using document classification and clustering [8] we can calculate the similarity of documents and potential document classifications (this means estimating relevant *DomainConcepts*); c) document locations: URLs, file names in file systems; d) document type: hypertext document versus E-Mail.

2. *Organizational Aspect*. The organizational aspect is derived from a so called organizational repository containing organizational entities and relations between them. For example, a person (*User*) is a *member* of the *Project* being *hosted-by* some *Department* being *part-of* some *Organization* and has *Interests* and so on (Tables 1,2). Workflow management systems, as well as, project management systems rely on such organizational modeling for sophisticated assignment of persons, roles, or alike to tasks. In contrast to the highly dynamic workflow/task modeling, this organizational model is quite static. Due to its static nature, it is possible to assign weights to the attributes and relations defined in this aspect.
3. *Behavioral Aspect*. Behavioral information is given mainly through a series of elementary actions performed by the user in the course of interaction with the operating system and applications. In the context ontology, they are classified into several categories, according to their nature, e.g., *AddBookmark*, as well as, *PrintDocument* are both *ArchiveOperations*. In other words, we modeled a taxonomy of *NativeOperations*. Whenever calculating the similarity of such elementary actions, this taxonomy is taken into account (*is-a* relations).
4. *Operational Aspect*. Operational information is acquired by observing active applications on the user's desktop at the moment. Typically used applications are classified into several categories according to their purpose.
5. *Causal Aspect*. Causal information conveys the goals and tasks of the user, as well as, the goal(s) of the tasks currently performed by the user. While the task's goals are annotated manually [2], the elicitation of the user's goals [1] depends on what information the user works with, which application she uses and how she uses them. It will be far from easy to elicit the user's current goal correctly, but it will deliver the most important evidence for estimating relevant workflow tasks.
6. *Chronological Aspect*. Chronological information is represented as temporal relations that bind certain instances in case the order of their addition to the task context matters. For example, chronological information provides ordering when comparing sets of behavioral events.

Also, temporal relations can contribute to the similarity between individual concepts. For example, if we compare the current user task contexts and the task contexts of other users and see identical *File* instances in all of them, we can assume that the context that has the most recent reference to this particular file is more similar to the current one.

4 Similarity Computation

As the task contexts are represented by complex information structures, it is necessary to employ similarity measures [12] for the computation of their similarity. To calculate the overall context similarity we perform the comparison of all its counterparts. We propose the following algorithm for contextual elements.

We represent ontology as a 2-tuple $O = \{C, R\}$, where C is the set of concepts, $R = \{\text{"is-a"}, \text{"part-of"}, \text{"followed-by"}\}$ – the set of possible relations between concepts.

We define task context as a 3-tuple $X = \{O, I, A\}$, where O is the ontology, I – is a set of instances, contextual elements ($i, i' \in I$), A is a set of attributes.

First, we want to determine, how much the two contextual elements have in common. The “is-a” similarity considers the generalization relations and shows how many superconcepts share the two concepts c and c' being compared. We use Jaccard [13] similarity measure for its calculation. So, to calculate the superconcepts similarity we define the set of concept’s superconcepts as

$$C_s(c_i) = \{c_j \in C : R(c_i, c_j) = \text{"is-a"} \vee c_i = c_j\} \quad (1)$$

$$sim_{is}(c, c') = \frac{|C_s(c) \cap C_s(c')|}{|C_s(c) \cup C_s(c')|}$$

As contextual elements have attributes, that are in turn other contextual elements or instances of primitive types related with a “part-of” relation, we compare these, too.

Since attributes are instances of different types, several attributes similarity calculation functions should be applied. Separate functions are provided for the primitive types, such as string, numbers and Boolean. Also, similarity functions are provided for complex entities, stored as instances of primitive types, for example, URL, path in a file system, e-mail address, room/phone number, etc. In case the attribute is an instance of another concept, we apply similarity measure formula $sim_t(i, i')$ recursively. In that case, we limit the depth of the recursion with some threshold, so that an infinite loop wouldn’t occur. The particular value of this threshold used by the implementation will depend on the preference of the user. We apply weights [10], assigned to the attributes here.

The “part-of” similarity is calculated as follows:

$$sim_{po}(i, i') = \frac{\sum_{j=1}^l \sum_{k=1}^m fsim_t(a_j^i, a_k^{i'}) * w_j}{l + m}, t \in T, a \in A \quad (2)$$

where $a_1^i \dots a_j^i, a_1^{i'} \dots a_k^{i'}$ – attributes of matching type and name in both instances being compared, l, m – numbers of attributes bound to each instance, $fsim_t$ is a function used for similarity calculation of attributes belonging to type t , $T = \{\text{"string"}, \text{"integer"}, \text{"boolean"}, \text{"path"}, \text{"URL"}, \text{"e-mail address"}\}$ or $sim_t(i, i')$; w_j – weight, assigned to the attribute indexed j .

To compare behavior information that is represented as a contiguous stream of elementary actions, we have to take into account the chronological ordering information, represented by the “followed-by” relations.

$$I_f(i_i) = \{i_j \in I : R(i_i, i_j) = \text{"followed-by"}\} \quad (3)$$

$$\text{sim}_{fb}(I_f(i), I_f(i')) = \frac{m}{n * s},$$

where m – overall number of matching instances in two event streams, n – sum of the numbers of events in each stream, s – number of matching sequences in both streams, $s=[1..n]$.

Given two contextual elements we calculate their object similarity [12] that is a global similarity measure aggregating the measures (1), (2) and (3)

$$\text{sim}_I(i, i') = \frac{\text{sim}_{is}(i, i') * w_{is} + \text{sim}_{po}(i, i') * w_{po} + \text{sim}_{fb}(i, i') * w_{fb}}{N} \quad (4)$$

where sim_{is} – “is-a” similarity [9], sim_{po} – “part-of” similarity (“Intra-Class Similarity” [12]), sim_{fb} – “followed-by” similarity, w_{is} – assigned weight of the “is-a” similarity, w_{po} – assigned weight of the “part-of” similarity, w_{fb} – assigned weight of the “followed-by” similarity, $N=[1;2;3]$ – N depends on how many similarity measures are applicable.

Having the similarity measure for contextual elements defined, we can compare task contexts as follows:

$$\text{sim}(X, X') = \frac{\sum_{j=1}^l \sum_{k=1}^m \text{sim}_I(i_l, i'_k)}{l * m} \quad (5)$$

where $(i_1..i_l)$ – contextual elements in context X , $(i'_1.. i'_m)$ contextual elements in context X' .

5 Implementation

The similarity between contextual elements is calculated and stored in the similarity table [11] dimensioned $N \times N$, where N is the overall number of contextual elements in all available contexts. As every contextual element is marked as belonging to one or several contexts, it is therefore possible to calculate the similarity between two contexts. The similarity measures are recalculated every time a contextual element is introduced into a context.

Figure 1 shows the process of context comparison. The user checks out a web-page, this event is captured by the user observation web-browser plug-in. A contextual element (CE), corresponding to the web-page is created and put into the user’s current task context. After that, this contextual element is compared with the contextual elements of existing task contexts. As it is known, what task context each contextual element belongs to, it’s possible to compute the similarity between task context and advice the user on which one is more relevant to his current task.

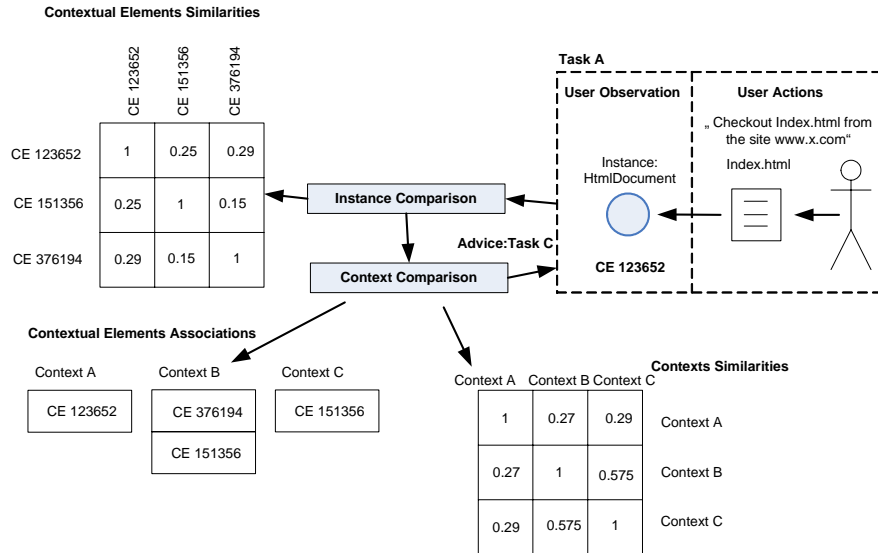


Fig. 1. Context Comparison Process

6 An example of the similarity measurement for task contexts

We would like to find workflow tasks relevant to the user's current context. We assume that a user has got 2 workflow tasks. They are as follows:

1. Writing a paper "A Similarity Measure for Task Contexts" for the Workshop at the Sixth International Conference on Case-Based Reasoning.
2. Writing an extended abstract "An explanatory component for decision support in the ecological monitoring of water resources" for the International Symposium on Explanation-aware Computing.

In our example the user is working at her PhD and therefore sends an email called "A question about CBR" to her scientific supervisor. Assume further that a document called "Explanation and knowledge-based systems" is stored as relevant information in the task "Writing an extended abstract" (Tables 1, 2). As we want to estimate whether that task could be relevant to the user's context, we compare both task contexts. This means, we calculate the similarity between the contextual elements in the user's context and the ones in the task context. In our example we therefore have to calculate, for instance, the similarity between the email and the document. First, we calculate the taxonomy ("is-a") similarity that shows how much the two elements have in common. Figure 2 shows the part of the information type ontology that we use to calculate the similarity between concepts "E-mail" and "Document". We see that $C_s(c) = \{\text{Email, Object, Thing}\}$ and $C_s(c') = \{\text{Document, File, Object, Thing}\}$. Thus, according to formula 1, $sim_{is}(c, c') = 2/5$. Then we calculate the instance ("part-of") similarity for these elements. To do it, we compare the attributes bound to them. They are as follows: $A(i) = \{\text{Sender} = \text{"Shkundina"}, \text{Receiver} = \text{"Roth-Berghofer"}\}$,

keyword="CBR", keyword="Paper", keyword="Explanation component", keyword="UML model"}, $A(i') = \{\text{Author}=\text{"Roth-Berghofer"}, \text{relevant D.C.}=\text{"Explanation"}\}$. Assuming the logical equality between the terms "sender" and "author" (in our case, it is true) and between "domain concept" and "keyword", we can see that there are only 2 matching attributes – relevant domain concept="Explanation" and keyword="Explanation component" of 7 attributes in sum (we do not consider the term "Receiver" as there are no similar notion related to the "Document" concept). The instance similarity thus equals to $2/7$ (we assume that the attributes are equally important and their corresponding weights are equal to 1). According to formula 4, the similarity between the two elements equals to $(2/5 + 2/7)/2$ (to make things easier in this example we assume that the instance and attribute similarities are of equal importance, so, the corresponding weights are equal to 1).

We calculate the similarity between the other elements in a similar manner. We assume that task contexts are similar when the similarity measure is more than 0.3. The similarities between the current user's context and the other available contexts were calculated and found less than 0.3. We therefore omit these calculations in the sake of brevity. According to formula 5, the similarity between the two task contexts mentioned above approximately equals to 0.3595. So, the task "Analyzing Decision Support Systems" is similar to the "Writing an extended abstract about explanatory component for decision support in the ecological monitoring of water resources". We can recommend the user to consider this context.

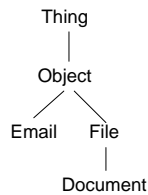


Fig. 2. A simplified fragment of the information ontology

Table 1. Contextual elements of the user's context

<i>Contextual Elements</i>	<i>Attributes</i>
<i>Documents</i>	
System databases and simulation models	<i>D. C.:</i> Ecology, Knowledge Bases
Decision support systems	<i>D. C.:</i> Ecology, IDSS <i>Author:</i> undefined
Modeling and other decision support tools	<i>D. C.:</i> Ecology, IDSS <i>Author:</i> Mr. Jeroen Kool
<i>E-mails</i>	
A question about CBR	<i>Sender:</i> Shkundina <i>Receiver:</i> Roth-Berghofer <i>Keywords:</i> CBR, Paper, Explanation component, UML model
<i>Users</i>	
Roth-Berghofer	<i>Interests:</i> Retrieval of Contexts, CBR, Explan. Syst., <i>Projects:</i> INRECA2
Shkundina	<i>Interests:</i> Workflow, Retrieval of Contexts, CBR, Explan. Syst.

Table 2. The contextual elements of the workflow task “Writing an extended abstract”

<i>Contextual Elements</i>		<i>Attributes</i>
<i>Documents</i>		
Explanation and knowledge-based systems		<i>D. C:</i> Explanation <i>Author:</i> Roth-Berghofer, T.
International symposium on explanation-aware systems		<i>D. C.:</i> Conferences, Explanation <i>Author:</i> undefined
Need for explanation component in ecological monitoring		<i>D. C.:</i> Ecology, Explanation <i>Author:</i> Shkundina
<i>E-mails</i>		
A question about explanation systems	<i>Sender:</i> Shkundina <i>Receiver:</i> Roth-Berghofer <i>Keywords:</i> Ecology, Wastewater treatment, Fuzzy logic, IDSS	
<i>Users</i>		
Roth-Berghofer	Interests: Retrieval and Modelling of Contexts, CBR, Explanation Systems, Philosophy and Informatics Projects: INRECA2	
Shkundina	Interests: Workflow, Retrieval of Contexts, CBR, Explan.Syst.	

7 Conclusion

We introduced an approach to the pro-active support of knowledge-workers during their daily routine based on building and comparing task contexts. The contexts of the workflow tasks and that of the user’s current task are compared. No generally accepted solution exists so far.

In the scope of our work we defined the notion of “task context”, an information structure, consisting of contextual elements that are instances of concepts, defined in the task context ontology. This makes possible grouping information of different origin into a single entity. Furthermore, the proposed similarity measure makes it possible to consider different types of attributes and relations. This allows for comparing the information structures known at the design time as well as for seamless extension by adding new types and defining new relations. The task context similarity calculation algorithm based on the proposed similarity measure was designed as a part of our organizational memory research project. It allows a user to find a task that is the most relevant to her current one and therefore benefit from reuse with minimum disturbance.

The further activity will be the technical implementation of this approach as a service for context sensitive assistance in EPOS. Performance estimation will be possible after the implementation of the prototype.

References

1. Sven Schwarz and Thomas Roth-Berghofer. Towards goal elicitation by user observation. In *Workshop Knowledge and Experience Management anlässlich des GI Fachgruppentreffens Wissensmanagement (FGWM 03), Karlsruhe*. GI, 2003.
2. Sven Schwarz. Task-Konzepte: Struktur und Semantik für Workflows. In Andreas Abecker, Rudi Studer, and York Sure, editors, *2. Konferenz Professionelles Wissensmanagement. Luzern, Schweiz*, number 28 in LNC. GI, Bonner Kollen-Verlag, 2003.
3. Andreas Abecker, Ansgar Bernardi, Heiko Maus, Michael Sintek, and Claudia Wenzel. Information supply for business processes – coupling workflow with document analysis and information retrieval. *Knowledge-Based Systems, Special Issue on AI in Knowledge Management*, 13(5):271–284, 2000.
4. Ludger van Elst, Felix-Robinson Ascho, Ansgar Bernardi, Heiko Maus, and Sven Schwarz. Weakly-structured workflows for knowledge-intensive tasks: An experimental evaluation. In *IEEE WETICE Workshop on Knowledge Management for Distributed Agile Processes: Models, Techniques, and Infrastructure (KMDAP03)*. IEEE Computer Press, 2003.
5. Ludger van Elst Andreas Abecker, Ansgar Bernardi. Agent technology for distributed organizational memories – the Frodo project. In *ICEIS 2003 – Artificial Intelligence and Decision Support Systems*, 2003.
6. Heiko Maus. Workflow context as a means for intelligent information support. In Varol Akman, Paolo Bouquet, R. Thomason, and R.A. Young, editors, *Modelling and Using Context. 3rd International and Interdisciplinary Conference, CONTEXT'01*, volume 2116 of LNAI. Springer, 2001.
7. Sven Schwarz. A context model for personal knowledge management. In *IJCAI 2005 – Second International Workshop on Modeling and Retrieval of Context*, to appear, 2005.
8. Heiko Maus, Harald Holz, Ansgar Bernardi, and Oleg Rostanin. A lightweight approach for proactive, task-specific information delivery. In *I-KNOW '05 – Special Track BPOKI'05 on Business Process Oriented Knowledge Infrastructures*. Springer, 2005.
9. Alexander Maedche and Valentin Zacharias. Clustering ontology-based metadata in the Semantic Web. In *European Conference on Principles of Data Mining and Knowledge Discovery (PKDD), Helsinki, Finland*, 2002 .
10. Mario Lenz. *Case retrieval nets as a model for building flexible information systems*. PhD thesis, Mathematisch-Naturwissenschaftliche Fakultät II der Humboldt-Universität zu Berlin, 1999.
11. Armin Stahl. *Learning of knowledge-intensive similarity-measures in case-based reasoning*. Ph. D. Thesis, Publisher dissertation.de, Volume 986, 2004.
12. Ralph Bergmann. *Experience Management: Foundations, Development Methodology, and Internet-Based Applications*, volume 2432 of LNAI, Springer, 2002.
13. A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Englewood Cliffs, N.J.: Prentice Hall, 1988.