Utilising Task-Patterns in Organisational Process Knowledge Sharing

Bo Hu[1], Ying Du[1], Liming Chen[2], Uwe V. Riss[1], Hans-Friedrich Witschel[1]

[1]SAP Research
{bo01.hu, ying.du, uwe.riss, hans-friedrich.witschel}@sap.com
[2]University of Ulster, UK
{l.chen}@ulster.ac.uk

Abstract Pattern based task management has been proposed as a promising approach to work experience reuse in knowledge intensive work environments. This paper inspects the need of organisational work experience sharing and reuse in the context of a real-life scenario based on use case studies. We developed a task pattern management system that supports process knowledge externalisation-internalisation. The system brings together task management related concepts and semantic technologies that materialise the former through a variety of semantic enhanced measures. Case studies were carried out for evaluating the proposed approach and also for drawing inspiration for future development.

1 Introduction

Recently *agility* has become an important requirement facilitating businesses of large as well as small sizes to reach their goals with reduced cost and increased efficiency. A major challenge presents in such a vision. The employees should effectively share their best practice in the form of process knowledge, also referred to as "know-how". Process knowledge, in many cases, manifests itself as the so-called tacit knowledge that is difficult to capture and denies easy reuse and sharing. In real-life, we acquire "know-how" through observation and gardened participation. Apprentices watched and learnt from their masters while working and achieving goals together. In the modern society, although apprenticeship still exists, its importance diminishes due to formal education system. There is, however, an evident shortcoming of modern education. That is while one can systematically acquire "know-what" in classroom or from online e-learning materials, "know-how" is more situated that is bound to particular problems and contexts and somehow proprietary to the individuals. It is not uncommon that when an experienced employee leaves an organisation, so does the "know-how" possessed and demonstrated by this employee. One way to ensure the sustainability of organisational "know-how" is to capture such knowledge and make it sharable and reusable. Moreover, by sharing knowledge of problem solving, we are able to extract and migrate such knowledge from individuals to teams and communities, and eventually to a stage that it becomes organisational knowlege. Externalising and formalising "know-how" is by no means a new research topic. Relevant research includes business process modelling, case-based reasoning, etc. Yet, the not-so flourishing results of current approaches are attributed to the rigidity of process models and the significant effort required in constructing such models. This daunting fact immediately prompted us to find a less formal approach facilitating process knowledge sharing. We propose to mimic what happens in real-life. That is demonstrating process knowledge through performing tasks (as *process knowledge externalisation*) and obtaining process knowledge through copying "actions" from accomplished tasks (referred to as *process knowledge internalisation*). During the externalisation phase, tacit process knowledge is concretised by attaching each task with pertinent information such as documents used, URLs visited, people contacted, etc. Accumulated data are abstracted and classified into repeatable patterns. Externalised process knowledge is not truly shared unless it is internalised again by others when similar tasks are to be fulfilled.

1.1 Motivating scenario: SAP ByDesign^{TM} Support

Nowadays, Digital Divide [12] manifests itself in a totally different form. Apart from being separated by our accessibility to the digital world (differed by how frequent and how well we use digital devices), we are more and more divided by how effectively we reuse and situate past experience into new problems and how proficiently we attach to it the relevant information. This is particularly evident when we studied the SAP ByDesign^{TM} Support Team where work efficiency is not solely determined by whether one has access to product documentation, manuals, internal Wiki pages, customer discussion forum—all the information is readily available to every employee and some is even open to public—rather by whether she knows how to leverage such information and project it upon the problem at hand. It is our contention that the difference in information handling capabilities leads to potentially significant variations in productivity, creativity, and work efficiency. When interviewing the support team in SAP, Galway, Ireland, we found significant difference between numbers of queries (referred to as incidents) handled by individual staff. An important issue raised was the lack of mechanism supporting reuse and sharing of past experiences, successful ones especially. All the interviewees were well trained and aware of the high-level formal procedures of dealing with incident reports. They, however, adopted different strategies when proceeding following the formal procedures. The difference in strategies, consequently, impinges significantly on the outcomes of work. For instance, some experts keep a separate record of key points when attending incidents (using notepad or software tools); some organise pertinent information of similar cases together in one file folder; some link together apparently different cases and emphasise on their commonality. Such good practice is not normally transferred from one expert to another and thus seldom go beyond individuals into team knowledge.

The need of sharing and reusing experience (in terms of established process knowledge) has been addressed previously in various contexts. This study, however, unveiled some technical barriers which, once overcome, might indeed change the terms of engagement. These barriers are: 1) the lack of low-cost formalisation of past experiences, 2) the lack of tools to systematically collect relevant information against a task, and 3) the lack of more sophisticated mechanisms for retrieving previous experiences.

1.2 Challenges and Considerations

Our user studies raised many challenges that can be, to a great extent, generalised to other situations where sharing and reuse of past problem solving experience is a major concern. Typically, good practices are formally modelled in workflow systems and made available to an organisation. However, workflow systems depend on predefined models, which are expensive and rigid. This is against our philosophy in searching for an answer to facilitating agility by cutting short the normal process formalisation cycle. We, therefore, focus on one easy-to-start approach to process knowledge sharing in terms of past problem solving experience. Process knowledge *internalisation* through task pattern reification provides an answer to our quest. Instead of formal process models, users adopt a task pattern and situate it with information specific to a problem's context leading to the instantiation and creation of new tasks from the task pattern. In summary, a task pattern is a carrier and externalisation of the user's past experiences. When the task pattern is applied as a template of creating new tasks, a knowledge worker internalises the past experiences embedded therein and contextualises them against her problems at hand (Figure 1).

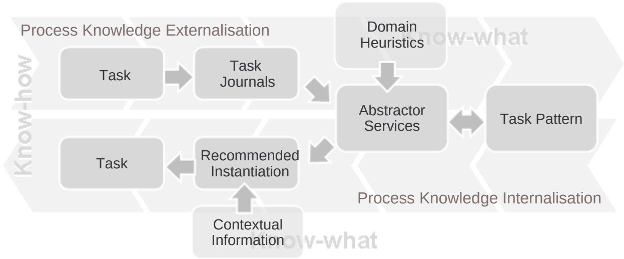


Figure 1. Process knowledge externalisation and internalisation

An onus of this process knowledge sharing duality is the additional effort and heavy user interaction in the task pattern lifecycle which have raised the major challenge of enabling the task pattern based approach. This is because knowledge workers are usually focusing on the tasks at hand and are likely to reject the extra work of abstracting existing tasks, e.g., contribute to task patterns, as part of their activities. Such activities merely add more administrative overhead for which the immediate benefit is not always clear. Hence, costs of performing such extra work must be reduced to as low as possible. A system enhanced with semantic technologies helped us to make a step forward in this direction.

In the follow, we first explain the pattern-based process knowledge sharing in Section 2 and how semantic technologies could be leveraged. In Section 3, we propose an architecture that facilitates the synergy of defined task patterns and bottom-up style task sharing. We detail what semantic technologies were employed. User study was carried out whose results are summarised in Section 4. Finally we conclude the paper in Section 5.

2 Sharing Process Knowledge as Task Patterns

The task pattern approach has been suggested in [9] as a low cost means to capture process knowledge in knowledge-intensive work. At the heart of this approach is the concept of dissecting process models on the level of individual tasks and use them to record and abstract activities necessary to fulfill the tasks. By doing so, the sharing of process knowledge becomes well focused and grounded as sharable and reusable patterns of tasks.

2.1 Task patterns

Hereinafter, we differentiate the concepts of *tasks*, *task patterns*, and *processes* as follows: A *process* is a collection of structured activities (tasks) with a precise goal to be achieved over a period of time. The activities (tasks) of a process are partially ordered and can be further divided into finer-grained sub-tasks. A *task* is an action requiring completion. A *task pattern* is an abstraction of tasks replacing specific resources with *abstractors*. *Task patterns* can be instantiated by assigning concrete instances to task resource abstractor as the abstraction of artefacts associated with tasks.

The Task Pattern approach bottom-uply involves users in the process management without implicating them in actual process management activities. This is done through task-oriented experience reuse or task copy transferring users' past experiences by copying details of accomplished and successful tasks. With task copying in mind, the next natural question is how such a copy operation will be realised. There are two options. First, entire task structures and details are duplicated, with the assumption that everything is implicitly relevant to the next task context; second, the user is responsible for explicitly selecting every detail to be copied. The former is likely to be useful for a small set of tasks. This is because it will likely result in information overload for a user. It results in a situation which requires the user to spend potentially more effort customising the duplicated task than to start a new one. The second option could also overwhelm users as it requires them to consider too many details from previous tasks, thus leading to a situation similar to that of the first. This reflection informs us that a more helpful position to consider the reuse of past experience lies somewhere in between these two extremes. To that end, we propose the concept of *task journal* as the basis on which previous experience can be shared (Figure 1). Task journals are the records of previous task activities and information artefacts, which are harvested by continually monitoring the interaction between users and a task management system, and collecting valuable information of events during task execution. The task history actually provides an explicit view on how the task is completed with critical information artefacts attached to it. The transition between task, task journal, and task pattern is fully supported in the task pattern management system (TPMS).

2.2 Supporting task pattern with semantic technologies

We investigated how existing semantic technologies are utilised in the context of a TPMS. We would argue that although semantic technologies are not the only solution to the challenges presenting in a TPMS, they offer unprecedent advantages over technologies that are not semantically enhanced. This can be seen from the followings aspects:

Firstly, semantic technologies offer machine processable meanings through ontologies. Even though obtaining shared domain knowledge increases the overhead, its value is evident in that i) a controlled vocabulary rooted in the ontology helps to regulate user interface reducing random inputs from users and thus increases system efficiency; ii) a common reference based on the ontology would serve as the foundation for aligning heterogeneous data; and iii) machine processable formalisation has a knock-on effect on automated reasoning. We are aware of arguments in the community against the formality of predefined global ontology due to its rigidity and the significant efforts involved therein. We, however, would like to emphasise that these would not be a major barrier in our application domain. Nowadays, organisations from different sectors, being both large enterprises and SMEs (Small and Medium Enterprises), have strict regulations in place and exercise an organisational common vocabulary to some extent as an effort towards organisational standards. Even though this is still far away from a formal ontology, it is already an embryonic form with which ontologies are considered as a natural subsequent step.

A formal ontology also demonstrates the capability of explicitly and implicitly linking apparently isolated data “islands”. Semantic technologies increase data linkage through well-formed logic formulae. Linking data together is important in process knowledge capturing. One goal of process knowledge sharing is to allow re-execution of the processes when and where it becomes necessary. In order to do so, it is crucial to not only share the skeleton problem solving steps but also “beef” each step up with necessary supporting evidences. For instance, when sharing the “flight ticket booking” process, backing each individual step with necessary organisational regulations, white papers, airline web pages, contacts of internal people in charge of payment, contacts of travel agencies, etc., would allow others to instantly pick up the correct process as well as *how* each step of it can be achieved with *what* means. These connections are not always evident and can only be established through instantiating the domain ontology with linkages among data concretising the properties defined in the ontology.

Tasks are introduced as instances of ontology concepts and properties, coded in RDF triples. In an organisation, archiving and easy retrieval of historical data is important for quality assurance purposes. This practical consideration leads to design requirements on the reliability and scalability of data repositories. Thus far, semantic data (mainly RDF triples) storage has been intensively studied. Efficient and scalable platforms include those of industrial strength (e.g. Oracle 11g, Jena SDB[1]) and those rooted in academic "proof-of-a-concept" prototypes (e.g. 3Store[2], Sesame[3]). The advantage of a purpose-built RDF triple store is its reasoning capability and the native query language facilitating easy access to apparently complex, intertwined RDF triples. In our scenario, RDF triple store is the backend for task instances, resource metadata, and semantic annotations.

RDF triples in its native representation suffer from poor readability. An intuitive user interface is preferable for better comprehension. In the meantime, the user interface should support team work, as in organisations collaboration sometimes is the key to fulfill tasks. We propose to materialise the frontend of data repository through Semantic Wikis[4]. Wiki is widely used as a platform for collecting and exchanging knowledge within communities. Collaboration is natural in Wikis in that registered users can jointly contribute to the contents of a topic while changes are managed with versioning tools. Enhanced with machine-processable markups, semantic wikis lend themselves to better organisation of information held therein. In this way the semantic structure is preserved while user interactions and information provision are better adapted to users' needs.

Finally, established semantic technologies can be of great help in addressing data interoperability. The so-called *semantic similarity* algorithms were recognised as an enabling technology for aligning heterogeneous perspectives over the same domain. Even though an organisation-wide ontology is reinforced, we still see the needs of attending data interoperability when one compares her goals against existing task pattern repository, retrieves established good practice and identifies candidate resources to be associated with / instantiate task patterns.

3 The Task Pattern Management System

The proposed TPMS (Figure 2) is underpinned by a task management ontology coded using OWL [2] as the ontology representation language. Having considered many existing approaches, we opted for the semantically enhanced wiki, Semantic Mediawiki (SMW) [5], for its improved content management, intuitive collaborative user interface, and smooth learning curve. Finally, we experimented and leveraged a variety of similarity algorithms to identify suitable tasks, task patterns, and information artefacts.

The system also consists of an shared repository for existing task patterns. Towards the aim of reusing past experience and work structures, knowledge

[1] http://jena.sourceforge.net/SDB/

[2] http://threestore.sourceforge.net/

[3] http://www.openrdf.org/

[4] http://semantic-mediawiki.org

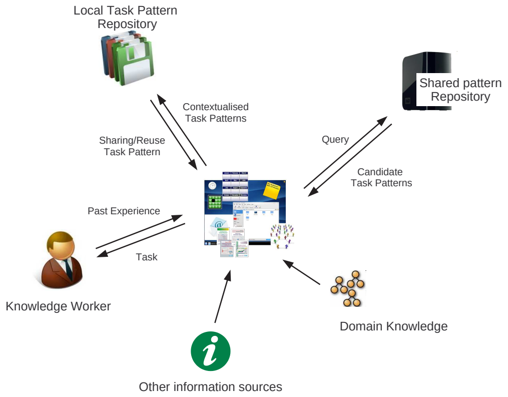

Local Task Pattern Repository
Contextualised Task Patterns
Sharing/Reuse Task Pattern
Shared pattern Repository
Query
Candidate Task Patterns
Past Experience
Task
Knowledge Worker
Domain Knowledge
Other information sources

Figure 2. System architecture

workers are unlikely to consider abstracting previous tasks to task patterns as an initial step. Therefore, we hypothesise that knowledge workers will refer to similar ongoing or completed tasks whose goals or contexts match those of her tasks at hand based on certain similarity measures [8], e.g. those reviewed in [7].

Since every task will differ in its context and details, the user is unlikely to be able to use the details directly from a task pattern. Instead, we expect that the user will be required to adapt existing task patterns to suit the context of the new task using domain knowledge encoded in the ontology and other useful information artefacts. For example, the user may select a document template as the basis for a report in the new task. Or the user may involve a person for the role suggested by the template task. This step of user adaptation through adding contextualised information and ontological domain knowledge is therefore necessary to enable the task patterns to be situated in the new work context. The user can then decide which adapted task patterns go into her private pattern repository and which are open to be shared with others. The adaption is effectively a classification and instantiation process during which the ontology and technologies built on top of ontologies play an important role.

3.1 Task Management Ontology, TMO

Ontology is currently considered the carrier of semantics. Its implication on our work is two-fold. On the one hand, ontology provides the necessary formalisation to increase data interoperability. When creating a task instance, one would impose some sort of structure on the information through ontologically regulated properties. For instance, one would normally annotate a person with names, affiliation, contact details, work places, etc. Such information facilitates both the readability of human users and accessibility of software tools. On the other hand, ontology offers inferencing capability. For instance, annotating artefacts based on the ontology enables us to perform (semi-)automated classification.

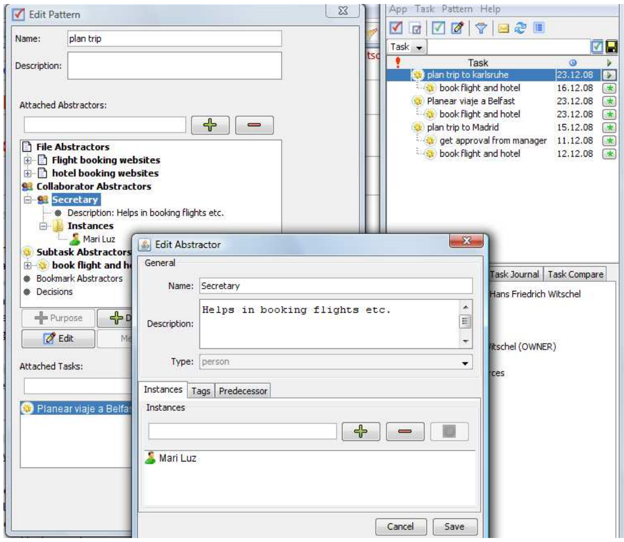
The central challenge of task management is providing effective task-related information support to knowledge workers. To this end, the TMO must be highly expressive and yet extensible to cater for ill-defined and continuously changing knowledge-intensive work situations. Consequently, the TMO is structured in two layers: (1) a set of concepts and resources which describe task-oriented information and work activities and (2) an underlying set of concepts which support the elaboration or concretisation of the more generic domain knowledge [10]. The first layer is centred around concept Task which is restricted with various properties, e.g. title, initiator, delegated_to (people), subtask, etc. The second layer includes such concepts as People, Document, Device, etc. These concepts are further refined with finer details. For instance, Document has WhitePaper, StaffManual, WebPage, etc, as sub-concepts. TMO is coded in OWL [2] due to its expressiveness, standard status, wide acceptance, and the Description Logic (DL [1]) ready feature for reasoning.

TMO was constructed by closely working together with target users, i.e. knowledge workers from various organisations. The evaluation of TMO was performed through expert review in an ethnographic study carried in the context of MATURE[5] and as part of the system evaluation to investigate whether the ontology facilitate a smooth integration of different functionalities of the TPMS.

3.2 Task Pattern Abstraction and Classification

Task journals play the role of experience carriers which convey various information during task execution. The decisive advantage of task journals is that they provide a chronological structure to task activities. This helps users to better understand which activities come first and which resources where used in the context of these activities. We utilise TMO to annotate and then classify and refine the information recorded in task journals. Whenever resources are attached to a task during its execution, a dialog box (Figure 3) is displayed to prompt users for annotating the resources with a list of concepts drawn from TMO. Annotated data are then processed with DL-based classification. For instance, when performing a task, one might make contact with a variety of individuals (introduced as instances of People) within different departments. Based on annotations detailing their positions and roles, these contacts are classified and abstracted as, for instance, "line manager", "cashier", "accountant", etc. Hence, who contributed to the fulfillment of previous tasks become less important as long as the correct types of people are contacted when one needs to carry out a similar task. Classifying information associated with tasks facilitates the creation of task patterns. Obtaining task patterns from task journals minimises users' effort to provide reusable process information to others and thus lowers one of the most important barriers in process knowledge sharing.

[5] www.mature-ip.eu



Edit Pattern

Name: plan trip

Description:

Attached Abstractors:

- File Abstractors
  - Flight booking websites
  - hotel booking websites
- Collaborator Abstractors
  - Secretary
    - Description: Helps in booking flights etc.
    - Instances
      - Mari Luz
- Subtask Abstractors
  - book flight and h
- Bookmark Abstractors
- Decisions

Attached Tasks:

- Planear viaje a Belfa

Task		
plan trip to karlsruhe	23.12.08	
book flight and hotel	16.12.08	
Planear viaje a Belfast	23.12.08	
book flight and hotel	23.12.08	
plan trip to Madrid	15.12.08	
get approval from manager	11.12.08	
book flight and hotel	12.12.08	

Hans Friedrich Witschel

itschel (OWNER)

Edit Abstractor

General

Name: Secretary

Description: Helps in booking flights etc.

Type: person

Instances

- Mari Luz

Figure 3. Task pattern editing environment

3.3 Semantic WiKi for content management

Our TPMS utilises SMW for a collaborative working environment. SMW extends the widely used MediaWiki[6] by adding machine-readable meta-data to a wiki article. A direct result of such an extension is improved search and classification, as well as better interoperability with other applications through importing existing ontologies and exporting contents in the standard OWL/RDF format, in addition to retaining the ease of collaboration that any wikis offer. Meanwhile, with an RDF interface, it is possible to query the wiki's content using popular query languages such as SPARQL. The advantage of such a combination is evident in the smooth learning curve for ordinary users while being sophisticated enough to create and store machine processable semantics.

Viewing in the light of task management, the decisive advantage of applying SMW is the combination of metadata and content handling with respect to task patterns. The metadata provided with a task pattern are ideal for automatic processing but less friendly for users to really understand what a task pattern is about. The SMW approach allows users to augment task patterns with sufficient textual descriptions that help users understand the goal and proceeding in a task pattern while at the same time support automatic manipulation.

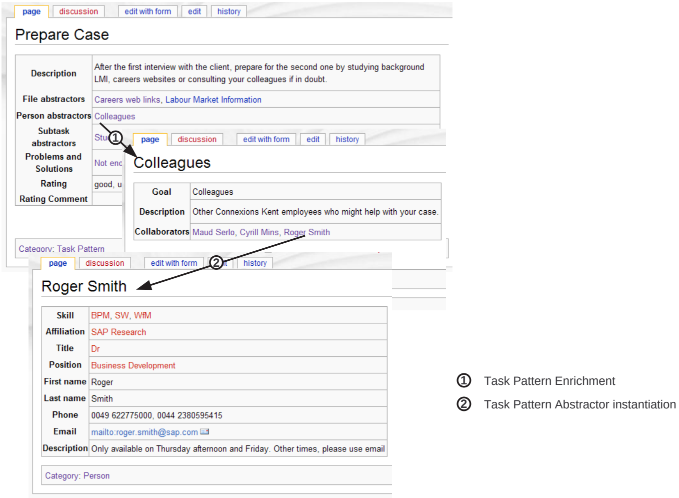
In the SMW, a task or task pattern is represented as an individual page with embedded inter-page links corresponding to the properties from an ontology (in

[6] semantic-mediawiki.org/wiki/MediaWiki

our case the TMO). We leverage the SMW templates for nicely-formatted task patterns (shown below).

```
{{Task
  |Description=                  |Document abstractors=
  |People abstractors=           |Subtask abstractors=
  |Problems and solutions=       |Rating=
  |Rating Comment=
}}
```

An exemplary "prepare case" task pattern is shown in Figure 4 which links the task pattern with two file abstractors, one person abstractor, one subtask, textual descriptions, etc. The person abstractor in turn restricts that only one's colleague can and should be contacted when carrying out case preparation. And finally, when one instantiates a task pattern, she links a person instance to the PersonAbstractor via Colleagues. The page Roger Smith instantiates Person concept in the ontology with concrete values for defined properties.



Prepare Case

Description	After the first interview with the client, prepare for the second one by studying background LMI, careers websites or consulting your colleagues if in doubt.
File abstractors	Careers web links, Labour Market Information
Person abstractors	Colleagues
Subtask abstractors	Stu
Problems and Solutions	Not enc
Rating	good, u
Rating Comment	

Category: Task Pattern

Colleagues

Goal	Colleagues
Description	Other Connexions Kent employees who might help with your case.
Collaborators	Maud Serlo, Cyrill Mins, Roger Smith

Roger Smith

Skill	BPM, SW, WfM
Affiliation	SAP Research
Title	Dr
Position	Business Development
First name	Roger
Last name	Smith
Phone	0049 622775000, 0044 2380595415
Email	mailto:roger.smith@sap.com
Description	Only available on Thursday afternoon and Friday. Other times, please use email

Category: Person

Figure 4. Instantiating task patterns in Semantic Wiki

As wikis have been widely used in different working environments, wiki-based TPMS can easily blend into the daily working environment of knowledge workers.

3.4 Similarity measures

One of the fundamental operations in our approach is to retrieve similar tasks that have been successfully carried out before. In our system, we utilise different similarity measures depending on the characteristics of the data.

String similarity The initial task patterns are normally retrieved by comparing task pattern descriptions with users' request, normally as a small set of keywords, e.g. "travel booking". Existing task patterns are summarised with plain text detailing the goal of the task pattern, features of its various abstractor services, outcomes of the tasks, and open issues. This is done utilising established techniques from Information Retrieval that the similarity of two pieces of textual descriptions as numeric values. The simplest form of text similarity is Edit Distance. Free Java based string distance library, SecondString [3], is used in our implementation.

Ontology-based similarity Pure string similarity algorithms ignore the structure of data which in many cases provide important information. They, therefore, are less attractive when the data in question are well structured against an ontology. For instance, when enriching a task pattern, users are prompted with candidate information artefacts that could instantiate abstractors. These artefacts are instances of People, WebPage, Document, etc, from the TMO. Hence, the way we understand artefacts is constrained by the ontology commitment. When information artefacts are properly annotated using the ontology, the comparison is tantamount to computing instance-level semantic similarity.

Instance-level semantic similarity is addressed in ontology mapping which is still an ongoing research. Many approaches have been proposed, implemented, and evaluated [7]. When comparing candidates, we reuse the algorithms developed in previous projects (e.g. CROSI [4]) that compute similarities of individual properties of task instances. The overall similarity is obtained from the similarities of individual properties. This is done by utilising weighted-average with human inputs indicating which property is more important than the others. For instance, one might emphasise on origins, dates, or places and thus give higher weights to these properties while others prefer the type and format of information artefacts. The domain heuristic knowledge was elicited from domain experts with an assumption that for a particular domain, e.g. ByDesign^{TM} Support Team, the importance of different attributes (as properties) of tasks should have been and can be clarified prior to running the similarity algorithm.

Graph-based similarity Similarity measure is needed for comparing tasks. A task consists of sub-tasks and is associated with artefacts, being human experts and information resources. This naturally becomes a labelled and directed graph, $G = (V, E)$ with nodes V corresponding to (sub-)tasks and individual artefacts and edges E corresponding to either part-whole relationships or associations. The easy conversion of task graphs inspires us to consider graph similarity measures. Graph similarity has been extensively studied (c.f. [6]).

In our TPMS, a task graph is forced into a task tree. The root of a task tree is the task itself. Children of the root are the first level sub-tasks which in turn have their sub-tasks as child nodes. The leaves of a *task-tree* are artefacts supporting the fulfillment of the task. We duplicate a piece of artefact when it is referred to by more than one sub-task. Focusing on task trees, we can largely ignore how a sub-task node is labelled. When two sub-tasks are supported by the same set of evidences, we can assume, based on the closed world assumption, they have overlapping instance data and require the same knowledge to proceed. This leads to a further assumption that sub-task requiring the same knowledge can be considered as similar tasks even though they are labelled differently. The closed world assumption is supported by two observations of our problem domain, i.e. process knowledge sharing in organisations. First, all the supporting evidences are shared and frequently used by a large number of employees and thus a common understanding can be easily negotiated. Second, such a set of artefacts is relatively stable. The creation or introduction of new information artefacts in a mature organisation is constrained by protocols and regulations. Hence, information artefacts that are not present are considered to be excluded from the organisational knowledge space and from our similarity computation. The algorithm is formalised as:

$$\texttt{sim}(t, t') = \frac{\mid \gamma(t) \cap \gamma(t') \mid}{\mid \gamma(t) \cup \gamma(t') \mid}$$

where $\gamma(x)$ gives the set of supporting artefacts of x and t and t' are two bottom-level subtasks.

For pairs of inner nodes from different task trees (T and T'), the similarity is computed from those of their children using tree edit distance [11]. Tree edit distance is an approximate measure computing the difference of two trees as a numeric value between 0 and 1. Based on the TreeDiff algorithm, task tree edit distance is defined as the minimum number of node deletes and inserts when one task tree is transformed into another.

$$\begin{aligned} \texttt{sim}(T, T') =& 1 - \texttt{diff}(T, T') \\ \texttt{diff}(T, T') =& \min \{\epsilon(S) \mid S \text{ is a sequence of edit operations } T \rightarrow T'\} \end{aligned}$$

where $\epsilon(\cdot)$ is the cost function mapping an edit operation to a numeric value based on users' preference. The initial alignment among nodes stems from the similarity among bottom-level subtask nodes computed as above.

4 Use case studies

The TPMS was studied in two different use scenarios. Thus far we mainly focused on i) the investigation of task-pattern as a general approach to process knowledge sharing and ii) the applicability of semantic technologies in task pattern management. In order to achieve the first goal, use case studies were carefully designed so that different types of organisations were presented. More specifically, one scenario was carried out in an organisation from the public sector while the other scenario focused on the support department of a large enterprise which regularly interacts with other departments within the same enterprise as well as external customers. In the meantime, the two scenarios differ in that the public sector organisation is normally in contact with individual clients with different types of requests while the support team's customers are relatively stable and predictable. The preliminary results are promising although further detailed studies are still necessary. The second goal was achieved through observing user behaviours so as to detect whether the use of semantic technologies distracts users from their ordinary working activities and whether extra work incurs.

Procedure These studies were performed as follows:

– Presentation of the prototype, its general functionality (including a very brief live demo) and the concept of task patterns.
– Pre-interview to determine how participants currently manage their tasks and how they deal with recurring tasks
– Performance phase during which participants had to work in two groups: the first group was asked to create a task, re-using information available in the system and at the same time refining that existing information into a task pattern. The second group was asked to create and populate another task by finding and using the task pattern provided by the first group.
– After the performance, participants were given a post-survey questionnaire to leave feedback and comments.
– Final discussion of the potentials of the task pattern concept in general.

Results We first examined the acceptance of the pattern-based approach to experience and process knowledge sharing. It was evident that all the interviewees acknowledge the significance of having a formal procedure to help them collect past experience when starting with a new task, while they pointed out that having the flexibility of attacking the problems in different manners is equally important. The interviewees in general saw task pattern as a low cost approach to facilitate process knowledge sharing, even though they have different working ethic, different regulations and are from different industrial sectors. Able to break down an apparent complex process into manageable task patterns was well received in both studies. During their daily work, they have both formal and less formal process prescribing how queries should be dealt with. It would be useful to deploy task pattern management system and then code some of these repeating queries as task patterns, thus making the sharing and reuse more flexible, allowing for individualised execution through process knowledge *internalisation*. Similar task patterns were actually identified across different industrial sectors, such as “report writing” and “travel booking”, suggesting a wide applicability of repeatable patterns in organisational process knowledge.

Enriching a task pattern by associating with it different resources is an important functionality that was identified as a welcoming feature of the prototype. A particular emphasis was made on the possibilities of using arbitrarily *any* artefacts to enrich a task pattern, as all of the interviewees frequently search internal competence management system for experts on particular subjects and subsequently talk to them over the phone or communicate via instant messaging. It is important to classify human experts in the same way as other information artefacts. To this end, semantic technologies (e.g. ontologies) were credited for making connections among data easier and more intuitive.

Usability was the most important thing raised unanimously by all the participants. During their fast-pace daily work, a software tool should be less interruptive and blend in with the other tools used. Wikis were accepted as a more user-friendly alternative as oppose to pop-up windows and dialog boxes. Some of the interviewees from the public sector were particular keen on a Wiki-based collaborative framework as such systems had already been embedded into their everyday working environment.

When asked for potential barriers, the participants would prefer the system to "speak" perfectly their "language" suggesting a certain terminology gap between the current version of TMO and some domain specific vocabulary. We, however, would argue that instead of denying the importance of having a task management ontology, this feedback emphasises the needs of mapping across different industrial sectors. Due to the relatively small size of TMO and the expressiveness of OWL, such a request can be easily accommodated by bridging new "jargons" with existing TMO concepts or properties, using either the *equivalent* construct from OWL or alternatively using SKOS[7] constructs.

Another problem raised in the post-interview discussion was that people have very different approaches to handle tasks (e.g. with respect to the granularity of modelling the task) and thus it might be hard to share patterns in some domains. Therefore, the application should be able to detect differences in granularity and react to it. On the other hand it is said that even if some abstractors were not reusable in a particular context, the pattern (with its subtasks) could be seen as a checklist and would always be useful in that respect for sharing useful experience in organisations.

5 Discussion and Conclusions

In this paper, we described a semantic-enriched, task pattern based approach to organisational process knowledge sharing and reuse. We view process knowledge as the "know-how" that are normally not transparent to others. In order to pass on to others, it goes through first a series of dissection and abstraction phases to externalise knowledge possessed by individuals into "tangible" and repeatable task patterns. We then employ a series of internalisation phases situating task patterns into the context of new tasks transferring others' experience to one's own. This operation is supported by tools and methods leveraging semantic technologies.

The presented user studies are the initial step of evaluating the applicability of task patterns. Though showing promising results, we acknowledge the major issues raised by users. One of such issues is the *incentive measure*. Like other systems relying on user inputs, our approach is based on the assumption that individuals are willing to record their problem solving activities and share such knowledge with others. Although this is still subject to further investigation, we would argue that motivating individuals within an organisation is less challenging. Employee performance evaluation is widely adopted in organisations from

[7] http://www.w3.org/2004/02/skos/

different sectors. The proposed framework is expected to increase the productivity of work and also help individual workers to identify gaps of her knowledge. It, therefore, presents good motivation for more active involvement in pattern-based good practice sharing and reuse.

While, obviously, there are many important issues to address, the crux of our immediate future work lies in the optimisation of the current TPMS and further improvement of the user interface. Thus far, our evaluation work focused on qualitative aspects investigating the underpinning theory of pattern-based process knowledge sharing and its impact in an organisational setting. Semantic technologies have indeed simplified the development and deployment. More evaluations are forthcoming to focus on quantitative usability aspects.

Acknowledgements

This work is supported under the MATURE IP funded by EU Framework 7.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
2. B. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuiness, P. Patel-Schneider, and L.A. Stein. *OWL Web Ontology Language Reference*. W3C, February 2004.
3. W. Cohen, P. Ravikumar, and S. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IIWeb*, pages 73–78, 2003.
4. Y. Kalfoglou and B. Hu. CROSI Mapping System (CMS) Results of the 2005 Ontology Alignment Contest. In *Proceedings of the 3rd International Conference on Knowledge Capture (KCap 2005) workshop on Integrating Ontologies, Banff, Canada*, October 2005.
5. M. Krötzsch, D. Vrandečić, and M. Völkel. Semantic mediawiki. In *Proceedings of the International Semantic Web Conference, ISWC2006*, pages 935–942, 2006.
6. L. Lovàsz and M. Plummer. *Matching Theory*. North-Holland, Amsterdam, 1986.
7. E. Rahm and P.A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, 2001.
8. U. V. Riss, U. Cress, J. Kimmerle, and S. Martin. Knowledge transfer by sharing task templates: two approaches and their psychological requirements. *Knowledge management Research & Practice*, (4):287–296, 2007.
9. U. V. Riss, A. Rickayzen, H. Maus, and W. M. P. van der Aalst. Challenges for business process and task management. *Journal of Universal Knowledge Management, Special Issue on Knowledge Infrastructures for the Support of Knowledge Intensive Business Processes*, pages 77–100, 2005.
10. Leo Sauermann, Ludger van Elst, and Andreas Dengel. Pimo - a framework for representing personal information models. In Tassilo Pellegrini and Sebastian Schaffert, editors, *Proceedings of I-Semantics' 07*, pages 270–277. JUCS, 2007.
11. J. Wang, K. Zhang, K. Jeong, and D. Shasha. A system for approximate tree matching. *IEEE Transactions on Knowledge Data Engineering*, 6(4):559–571, 1994.
12. K. Williams. What is the digital divide? Technical report.