2007

A Hybrid Method for E-Process Selection

Frina Albertyn
Eastern Institute of Technology,Napier, falbertyn@eit.ac.nz

Roland Kaschek
Massey University, R.H.Kaschek@massey.ac.nz

A Hybrid Method for E-Process Selection

Frina Albertyn
Information Technology Section, Faculty of Business and Computing
Eastern Institute of Technology
Napier, New Zealand
Email: falbertyn@eit.ac.nz

Roland Kaschek
Department of Information Systems
Massey University
New Zealand
Email: R.H.Kaschek@massey.ac.nz

Abstract

A number of e-Processes (i.e. software processes for developing e-Commerce information systems) are available in industry. It is difficult to select the best suited e-Process for a case at hand. At the same time this selection is important because functionality and quality of any system under development will depend on the instantiated software process. The knowledge required for the selection task cannot be easily realized. That task can be considered as an instance of multi attribute decision making and several of the attributes to consider are likely to conflict with each other. An efficient and effective approach is needed to selecting software processes for developing e-commerce systems. In this paper we propose such an approach. It is hybrid as it rests on case-based reasoning, multi attribute decision making, and social choice methods. To demonstrate how our approach works we briefly discuss a case study.

Keywords

ECIS, e-Process, e-commerce development, case-based reasoning, decision making

Introduction

E-Commerce can be defined as the process of doing real-time business transactions via telecommunication networks. Customer and supplier are usually located in different places and in general no agreement has been made between the cooperating parties prior to the transaction taking place (Mattila 1998). E-Commerce is a broad concept that includes viewing goods on sale over the internet, assessing the good quality and suitability, selecting goods for purchase, and performing purchase transactions using one of a variety of payment methods.

Developers of e-Commerce information systems (eCIS) are under pressure to develop applications quickly, i.e., within the scheduled time-frame. The widespread lack of process discipline and procedural guidance available for such market driven environments highlights the need for a framework to support practitioners' development efforts (Antón, Carter et al. 2001).

(Gruhn and Schope 2002) point out a number of impact factors of eCIS development that are specific to this class of systems. Among these are the wide variety of users interacting with the eCIS, the volatility of that user group, the typically high degree of interaction with other information systems, and the condition that these systems not necessarily are administered by the same authority. Clearly the first three of these points increase the importance of the user interface design. The fourth point makes designing and implementing the system functionality more demanding and system maintenance more difficult.

A software process is a framework or reference model for the development of computer applications such as information systems. In terms of selecting software processes three key impact factors are frequently mentioned in the literature, i.e. productivity, quality, and case specifics (Brynjolfsson 2003; MacCormack, Kemerer et al. 2003; Mansar, Reijers et al. 2005). The efficiency of software development (i.e. producing high quality at a low price) to some degree can be controlled based on software process quantification (Dumke and Foltin 1996). Often software process selection employs a listing of software process features. It is also important to address software quality features such as efficiency, maintainability, portability etc (Dumke and Foltin 1996).

A number of research papers have focused on method selection within several domains. Kaschek and Mayr in (Kaschek and Mayr 1996) and (Kaschek and Mayr 1998) proposed frameworks for comparing object-oriented modelling (OOM) tools and analysis methods, respectively. Using these frameworks, they built two detailed lists of quality aspects for selecting both tools and methods. That work in its time was innovative with respect to the ontology provided. It, however, was conservative with respect to assessing ontology-conform items, i.e., OOM modelling tools or analysis methods. They used value-benefit-analysis (VBA) for scoring these items. VBA, however, is (and was) known to be weak with respect to justification of the used part-scores and the weights that in a weighted sum approach are used to obtain the individual scores that finally are used for ranking the eligible items. A recent paper critiquing VBA on this ground is (Bernroider and Mitlöhner 2007). VBA as a selection method is also weak because of its lack of method support regarding conducting trade-off and sensitivity analysis.

In earlier work we have first adapted the mentioned ontology to our task of selecting e-CIS software processes Albertyn (Albertyn 2005, 2007; Albertyn and Kaschek 2004, 2005; Albertyn and Zlatkin 2004). We refer to such a software process as e-Process. We have also started to adapt more appropriate scoring methods such as the Analytical Hierarchy Process (AHP) (Saaty, 1990) that is used in multi-attribute-decision-making (MADM) and that in the employed version is based on pair-wise rankings of the alternatives to choose from as well as the quality aspects to be considered in the case at hand and that are taken from our ontology (Kaschek et al. 2006, Kaschek et al. 2007). In (Kaschek et al. 2007) we have put some emphasis on tradeoff analysis and sensitivity analysis of e-Process selection in case of AHP being used as the main decision making tool. Our e-Process ontology consists of a two-level hierarchy of e-Process characteristics. Al-Humaidan and Rossiter did similar work by developing a taxonomy for classifying methodologies of workflow-oriented information systems (Al-Humaidan and Rossiter 2000).

We presuppose that an approach to e-Process selection must meet a number of characteristics: (1) it needs to be light weighted and such that it can be applied in smaller organizations and by staff without expertise in e-Process selection methodology; (2) it should aid in acquiring e-Process selection expertise; (3) it should employ a step-wise procedure that occurs as natural to involved staff; and (4) it should be scalable with respect to the number of e-Process selection cases that staff have to deal with (Albertyn 2005, 2007; Albertyn and Kaschek 2004, 2005).

We have been researching different methods for selecting e-Processes. In this paper the candidate areas from which we are going to reuse method chunks are MADM, social choice methods (SCM), and case-based reasoning (CBR). From a voter's point of view the main difference between AHP (Saaty, 1990) and SCM is that AHP allows for the use of several degrees of preference of one alternative over another while SCM only permits using one such degree. CBR has been used successfully in various domains such as finance, medical, diagnostic, repair manuals, helpdesk assistance (Macaya, Melendez et al. 2002). It is heavily related to BVA, as will be seen below.

Considering the basic ways (employing experts, using rules, or a case-base) of incorporating the required knowledge into an e-Process selection task uncovers drawbacks. For example, suitable experts may be difficult to identify and expensive to hire. Furthermore, the employed rules well may lead to inconclusive results or may turn out to be inappropriate generalizations for the case at hand. Certainly each of CBR, MADM, and SCM has its weaknesses. We, however, think that by combining them some of these can be overcome. An advantage of CBR is that reusing problem solutions of past cases similar to the one at hand may reduce the need for green-field approaches to the analysis of that case at hand (Cunningham and Bonzano 1999). SCM are easily applied, as they are based on obtaining pair-wise rankings of eligible items. The AHP in MADM is a very well respected theory. Obviously computer support is needed for these methods being applied comfortably and its implementation has been started.

For a software developer it is not necessarily the most natural thing to -prior to starting the development- think about the best software process to use in the case at hand. Still many developers have also a limited knowledge of different software processes. The working conditions not necessarily permit them to regularly refresh their method knowledge. In some sense therefore our research is of limited impact. However, for a consultant or a department head the condition may be rather different in particular in cases that involve larger software vendors or organizations that have many developer staff. The latter very well may be the case with large international organizations such as banks, insurance companies, or vehicle manufacturers. For those in a case at hand, very well a choice may be possible of the software process to be followed. Our approach not only can help making an appropriate decision in a case at hand. It also can contribute to identifying a qualification need with respect to software processes.

Paper outline. In the next section the fundamentals of our method is discussed and then the specific aspects of the method is discussed in the section thereafter. We then have the concise representation of our method in a step-wise fashion and apply our method to an actual selection problem. Finally in the last section we conclude the paper with some closing comments and a discussion on future research direction.

The Fundamentals of our E-Process Selection Method

There is a number of guidelines and standards in place for the assessment and selection of software processes e.g. ISO 15504, also known as Software Process Improvement Capability Determination (SPICE), which is a "framework for the assessment of software processes". This standard is aimed at setting out a clear model for process comparison. It models processes to manage, control, guide and monitor software development. This model is then used to measure what a development organization or project team actually does during software development. This information is analyzed to identify weaknesses and drive improvement. It also identifies strengths that can be continued or integrated into common practice for that organization or team (Loon 2005; Wikipedia 2007). To the best of our knowledge, however, no generally accepted theory of e-Process selection exists. It is not even commonly accepted that such a method should be used and that its use can be economically viable. We have decided to combine CBR, MADM, and SCM for defining a light weight method for e-Process selection that actually can be used in practice.

CBR is an established artificial intelligence paradigm (Leake 1996a). It is a technique used for organizing and aiding with computer means the human reasoning process which is based on experiences. Experience based reasoning is often understood as based on humans mentally searching for past situations that are similar to present ones and reusing the experience gained from dealing with those past situations (Leake 1996a; Mansar, Marir et al. 2003). Leake (p. 2) lists a number of works that provide empirical evidence for a relevant part of human reasoning to be experience based rather than being rule-based. Many test persons preferred solving specific problems based on similar experiences that they made in the past rather than on preparing chains of deductions involving generalized knowledge and abstract reasoning rules.

CBR is appropriate for tasks for the solution of which no generally accepted theories exist. CBR aids experts in recording, organizing, storing, and retrieving past decision-making cases and benefiting from that technology-supported and organized form of memorizing. A database, the so-called case-base is developed and on top of the cases a similarity measure defined (Inductive Solutions 2006). That similarity measure is used for selecting from the case-base a number of cases that are highly similar to the case at hand. According to (Leake 1996a) CBR rests on the assumptions that (1) very similar problems admit very similar solutions; and that (2) problems tend to recur if one does not change significantly ones life or living conditions.

CBR is inductive, as it provides new semantic information based on obtaining quantified similarity of cases. Deductive problem solving approaches are based on analysing available information with logical means and transforming that information into a form suitable for decision making (Cunningham and Bonzano 1999; Inductive Solutions 2006). Because of its inductive nature CBR can cope with irregularities in problem definitions better than deductive approaches can. In particular no checking of consistency of new cases with respect to old cases is needed (Inductive Solutions 2006). We therefore consider CBR as suitable for aiding in e-Process selection.

A case includes a problem, a solution, and an outcome, i.e., the result of applying the solution (including solution assessment), and a context, i.e., information about conditions in which the problem occurs and how to apply the solution (Watson and Marir 1994). The unit of problem, solution, and outcome is also known as case content. For our purpose the solution application information can be ignored because a mature software process is including this kind of information.

The four basic steps defined for CBR are (Watson and Marir 1994; Cunningham and Bonzano 1999): (1) retrieve a number of most similar cases; (2) reuse the cases to attempt to solve the problem; (3) revise the proposed solution if necessary; and (4) retain the new solution as a part of a new case. We, however, do not follow exactly that procedure. Rather we opt for the retrieve-propose schema mentioned by (Leake, 1996a) because we do not want to encourage software developers to create their own software processes. Also, as Leake observes, in CBR for solution adaptation one tends to rely on adaptation rules. The existence of such rules seems to presuppose the existence of a theory of the problem domain. However, one tends to employ CBR specifically if one does not know such a theory. Thus we do not particularly favour the solution adaptation in CBR. In recognition of the need to provide an efficient approach we suggest that developers first make up their mind about the e-Process to use in the case at hand. Then from the case-base we suggest retrieving a number of recent cases in which the developer favoured e-Process has been used. If the developers then decide that these cases are sufficiently similar to the case at hand and in these cases the favoured alternative has been considered as a good solution then we suggest just going along with that favoured solution. Otherwise we suggest the retrieve-proposing schema being started.

Adapting the recommendation of (Aamodt and Plaza 1996) we presuppose the retrieval to be carried out by first identifying the relevant features, performing a pre-selection of cases from the case-base after that, and finally selecting the one among these that matches best the case at hand. We recommend identifying the features that are considered as important for the case at hand by using instances of SCM such as the Borda rule or the Kemeny rule, because these can be computed from the margins alone. For two e-Processes P, P' the margin *m(P*

, P') of *P* versus *P'* is the number of times *P* is preferred over *P'* in pair-wise comparisons minus the number of times *P'* is preferred over *P* in these comparisons. See for more detail (Bernroider and Mitlöhner 2007) and the references given there. We are aware of attacks at decision making processes that employ the Borda rule. However, we do not admit proposing new e-Processes in a case of e-Process selection and think that this suffices in our case as we don't see any stakeholder benefiting from a particular e-Process being chosen.

We presuppose that SCM is applied such that admissible items (i.e., e-Processes) are ranked pair-wise by each of the involved staff (may these be dedicated experts, project leaders, developers, or similar). SCM knows a number of different ways of obtaining an integrated ranking based on such a list of pair-wise rankings. We presuppose that the involved staff may discuss things with each other prior to the pair-wise rankings be performed. They, however, do these rankings in private and once the e-Process selection task has been initiated no new e-Processes may be proposed for use in the case at hand. We think that this way several ways of attacking the instantiated SCM approach can be prevented from happening. We also presuppose that there are not many incentives in an organization that might lead to such attacks being launched. It is, however, obvious that more experiments need to be made so we can better understand the conditions under which SCM approaches work best. Note, that in (Bernroider and Mitlöhner 2007) it is shown that with each one of a number of SCM approaches one could reproduce in two different studies the outcomes produced by BVA.

When building the knowledge base a number of methods can be used to organise, retrieve, utilise and index the past cases (Harrison 1997). According to Cunningham (Cunningham and Bonzano 1999) the tasks involved with building up the knowledge base are identification of the "real world" problem and representation the key components thereof in the knowledge base. Next it is important to develop the inference mechanism that describes the causal interactions involved in deriving solutions. The inference mechanism is implemented using the knowledge base with the cases of solved problems and a mechanism to retrieve and adapt these cases (Cunningham and Bonzano 1999).

Richter (Richter 1998) has identified four so-called knowledge containers. These are aspects of CBR approaches by means of which knowledge can be represented in the knowledge base. These are (1) the vocabulary used; (2) the similarity measure; (3) the case-base; and (4) the solution transformation. The CBR community has widely accepted these as a natural organisation of knowledge. Since we do not employ the solution transformation we are only left with the first three of these containers. For now we do not see how we additionally to aiming at standardizing the vocabulary used –at the method level- could exploit that vocabulary. In our method we thus focus on the case-base and the similarity measure.

Specific Aspects of our Selection Method

We adapt now Richter's knowledge containers to our problem, i.e. e-Process selection.

Similarity Measure

We note that social choice approaches or the AHP could be used for obtaining similarity rankings after a heuristic pre-selection has been conducted. It is, however, common to obtain similarity measures automatically. Computing cost would tend to be too high if one would obtain the similarity to the case at hand of all cases in the case-base. Therefore heuristic rules for pre-selecting cases are in use. However, currently our case-base is still small and also due to the experimental character of our research all similarities will be obtained. According to (Coyle, Doyle et. al. 2004) the similarity between two cases Q and C is defined as the sum of the weighted similarity of the cases' constituent features:

$$Sim(Q,C) = \sum_{f \in F} w_f * \sigma_f(q_f, c_f).$$

In this equation w_f is the constituent feature weight, σ_f the similarity measure applied to feature *f* of Q and C, and *F* the set of all features. The weights are seen as feature attributes. The similarity measures obviously are more complex. (Coyle, Doyle et. al. 2004) use three different kinds of feature similarity measures. These are (1) the exact similarity measure, i.e., the similarity score is 1 if the feature values are equal and is 0 otherwise; (2) difference based similarity measure, i.e., the similarity score depends on the difference of the numerical feature values but not necessarily is 0 for non-equal feature vales; and (3) complex similarities, i.e., all other similarity measures. Using the difference based similarity measure essentially turns CBR into a version of BVA. In our method SCM also plays a role. We think the weaknesses of BVA (i.e. that it often is very hard to score items on a scale according to a number of features) can be overcome by the incorporation of SCM method parts that only rely on ranking items.

In selecting an E-Process both measure kinds (1) or (2) could be used. We are going to use (2), i.e., the difference based similarity measure. Currently we obtain values of weights and feature similarities from the developers who were asked to assess these variables quantitatively. In future we plan to account for the well-known critique of this approach by using SCM or AHP for obtaining scores based on rankings provided by developers. Prior to using these methods we need to finish implementing the software systems dedicated to aid humans in applying these methods.

Case-base

The quality of e-Processes can be seen as being multi-faceted, as is often experienced for other complex entities such as information systems (Ghezzi, C., Jazyeri, M., et al 2004). Individual quality aspects, however, are often considered as too broad and unspecific and are therefore decomposed into lists of lower level quality aspects. We exploit a two level system of e-Process quality aspects.

Using both the VBA (Albertyn 2005) and the AHP (Albertyn, Kaschek et al. 2007) a case-base has been developed. It has a number of levels that are used in the comparison. First there is information about each of the projects/cases being stored. Then for each of the cases we have group values (high-level quality aspects) that group a number of characteristics (Lower level quality aspects). Then we have weights assigned to each of the characteristics.

Since we focus on CBR as applied for e-Process selection and because similar systems of quality aspects have been used before for similar purposes we do not specifically argue for the quality aspects used in this study. They are, however, loosely based on (Kaschek, R., Pavlov, et. al. 2006) and have been used in (Albertyn 2005, 2007; Albertyn and Kaschek 2004, 2005). Our quality aspects are:[1]

- **e-Process characteristics**, i.e. the modelling notions, abstraction concepts, and other aid suggested by the e-Process
    - *Completeness*, i.e. the degree to which the e-Process provides means of expression (such as modelling notions, abstraction concepts, patterns, and anti-patterns) that enable the ECIS developer to effectively and efficiently solve development tasks within the domain of application of the e-Process.
    - *Understandability*, i.e. the degree to which one easily understands the e-Process.
    - *Visibility*, i.e. the degree to which the defined activities of the e-Process result in clear outcomes and effective and efficient project management is enabled.
    - *Supportability*, i.e. the degree to which CASE tools are accessible that aid in using the e-Process.
    - *Maintainability*, i.e. the degree to which the e-Process enables managing requirements in particular with respect to change requests after project commencement.
- **Specific Quality concepts of the e-Process** (its reliability, robustness etc)
    - *Readability*, i.e. the degree to which the notation prescribed for use in the e-Process is easy to read.
    - *Reliability*, i.e. the degree to which the e-Process is designed in such a way that errors in the development process are avoided or identified and fixed prior to system deployment.
    - *Robustness*, i.e. the degree to which the e-Process continues to aid developers in case of unexpected events occurring.
- **Cost** for using the e-Process
    - *Development Budget*, i.e. the degree to which a tight development budget does not tend to reduce the applicability of the e-Process;
    - *Running Costs*, i.e. the degree to which the ECIS developed with the e-Process will run according to allowed budget.
- **Domain impact**, i.e. the impact of the project domain
    - *Infrastructure*, i.e. the degree to which the technical environment of the enterprise affects the applicability of the e-Process.
    - *Enterprise Culture*, i.e. the degree to which matching the e-Process to the enterprise culture does not reduce the applicability of this e-Process.

[1] Top-level and bottom-level aspects are indicated by bold facing and italicizing respectively. The latter aspects are indented.

- *Technology*, i.e. the degree to which matching the e-Process to all other technology being used in the enterprise does not reduce the applicability of this e-Process.
- *Geographic Interaction*, i.e. the degree to which the globalization of the enterprise is necessary for the e-Process being applied.
- *IT Strategy*, i.e. the degree to which the enterprise's system development standards (specifically ECIS development) do not reduce the applicability of this e-Process.
- *Business Strategy*, i.e. the degree to which the enterprise's business strategy does not reduce the applicability of this e-Process.
- *Team Experience*, i.e. the degree to which the involvement of the team with previous ECIS development affects the applicability of this e-Process.
- *Domain Knowledge*, i.e. the degree to which the domain knowledge of the development team affects the applicability of this e-Process.
- *E-Process Knowledge*, i.e. the degree to which the knowledge of the e-Process of the development team affects the applicability of this e-Process.
- *Development Time*, i.e. the degree to which the development time influences the applicability of this e-Process.

- **Usability**, i.e. the e-Process aid in developing a high quality ECIS
  - *Functionality*, i.e. the degree to which the e-Process specifies the required development artefacts (such as requirements, design, implementation, testing, etc.) and instructs how to create and use them.
  - *Manageability*, i.e. the degree to which the e-Process aids managing projects effectively (including planning, tracking, and risk management etc.).
  - *Quality assurance*, i.e. the degree to which the e-Process aids developers in following the principles, requirements, and recommendations of the Total Quality Management announced at ISO 9000:2000, ISO 9001 and ISO 9004.
  - *Adjustability*, i.e. the degree to which the e-Process can be adjusted to meet the specific needs of the ECIS project in the most effective way possible.
- **Compatibility** of the e-Process with other methodologies
  - *Exchangeability*, i.e. the degree to which the artefacts of the given e-Process can be exchanged between the tools implementing this e-Process.
  - *Map ability*, i.e. the degree to which the artefacts of the given e-Process can be mapped into the artefacts of other e-Processes.
- **Maturity**, i.e. e-Process stability, tool support, documentation etc.
  - *Stability*, i.e. the degree to which the e-Process has been proven, i.e., its standing.
  - *Tool support*, i.e. the degree of availability of the tools supporting this e-Process (such as version control and document/workflow management systems) and the quality of these tools.
  - *Documentation*, i.e. the degree to which adequate documentation is enabled.

Concise Presentation of our Method

The case-base was developed and implemented during our earlier research on e-Process selection but in order to keep the paper short this is ignored here. That implementation is part of a project to implement a tool-box that aids e-Process selection and which provides a number of respective methods.

To sum up, our method, represented in a step-wise fashion, consists in:

1. Use an SCM for selecting an e-Process most suitable for the case at hand. Verify that selection with inspecting some of the latest cases for which that selection was applied. Take into consideration the case-context information as well as the solution assessment.
2. If the selection made in step 1 can be verified, then go along with it. Otherwise:

   2.1 Use an SCM approach to identify the feature set to be used.

2.2 Rank the features (according to whether no preference, light preference, moderate preference, strong preference, or extreme preference is given to one feature over another one) in a pair-wise fashion and use the AHP to obtain feature weights.

2.3 Score the case at hand with respect to the features selected. A light weight method such as an SCM should be used for this.

2.4. Choose a lower boundary b(Q) for limiting the similarity of case C in the case base to case at hand Q, i.e., b(Q) ≤ Sim(Q,C), for preselecting those cases C that are considered to match best the case at hand Q. Perform that preselection.

2.5 Use an SCM for identifying the one case in the pre-selection that is considered to be the one most similar to the case at hand and use the e-Process used in that case.

3. Store in the case-base the new case together with the solution as well as a brief solution assessment (obtained post project completion).

That procedure is efficient, as in case expertise emerges the more complex operations are not even executed. Rather the initial choice of the involved staff will be used. In case that expertise is not yet developed enough so that a selection cannot be made initially that would be consistent with past experiences the procedure continues with prescribing more advanced techniques. The procedure also helps acquiring expertise as it aids in recording, organizing and reusing cases of e-Process selection. We consider our method as natural since we pointed out evidence for case-based reasoning being applied in problem solving by many test persons. Varying the boundary b(Q) allows for tuning the cardinality of the preselected set of past cases. So our method is scalable.

We now apply our method to an example problem for exemplifying how it works. We aim to determine a best-suited e-Process available for the development of the eCIS for the problem by using recorded experience. We first briefly describe the problem.

The Example Problem

A New Zealand packaging company has asked their Information Systems department to set up an e-Commerce site, to allow their customers to place orders online. This company specializes in cardboard packaging. Their customers are based nation wide and are mostly large manufacturers.

Their Information System department is currently running their website and maintains the Information Systems required in the plant. The IT personnel consist of 5 people, 3 of which have development experience. Only one of the three developers is experienced in website/ECIS development, all three developers have been using development processes to support most of their development. All three developers will be involved in the ECIS development process.

Table 1: Comparing top level quality aspects of the problem to two cases

Top Level Quality Aspect	Weight of quality aspect	Example problem	Case Base example 1	Similarity value (Case 1)	Case Base example 2	Similarity value (Case 2)
	A	B	C	A × |B-C|	D	A × |B-D|
e-Process aspects	0.20	0,300	0,100	0,040	0,143	0.031
Quality concepts	0.22	0,048	0,148	0,022	0,213	0,036
Cost	0.3	0,150	0,250	0,030	0,251	0,030
Domain impact	0.1	0,143	0,040	0,010	0,204	0,006
Usability	0.04	0,238	0,338	0.004	0,123	0,005
Compatibility	0.05	0,048	0,050	0.000	0,022	0,001
Maturity	0.09	0,073	0,074	0,000	0,062	0,001
				0,106		0,110

Method Application

As our case base is still very small all the stored cases are evaluated for similarity. We will however just show two cases in order to demonstrate our process. In the first case-base example the resulting e-Process recommended was Storyboarding and User Profiling and in the second case-base example the resulting e-Process recommended was the Rational Unified Process.

In table 1 the values provided by the developers of both the problem at hand as well as the developer information for the two historical eCIS developments are shown. This table also provides the weights that experts provided for each of the top level quality aspects. The Case Base currently only includes historical information on e-Process choices for eCIS using one of the following four software processes:

- Rational Unified Process (RUP);
- Open source development process – the “bazaar” approach (OSS);
- Agile and extreme programming ECIS development process (AX);
- Development process using storyboarding and user profiling (SBUP).

These four software processes were chosen because they represent a broad spectrum of acceptable software processes. The system of e-Process quality aspects given in the previous section was used.

Table 2: Low level quality assessment results

Quality Aspect	Weight of quality aspect	Example problem	Case Base example 1	Similarity value (Case 1)	Case Base example 2	Similarity value (Case 2)
Completeness	1	0,60	0,12	0,48	0,20	0,40
Understandability	1	0,17	0,50	0,33	0,17	0,00
Visibility	1	0,58	0,12	0,46	0,19	0,39
Supportability	1	0,60	0,12	0,48	0,20	0,40
Maintainability	1	0,19	0,19	0,00	0,56	0,37
Readability	1	0,54	0,18	0,36	0,18	0,36
Reliability	1	0,19	0,19	0,00	0,56	0,37
Robustness	1	0,19	0,56	0,37	0,19	0,00
Development Budget	1	0,13	0,38	0,25	0,38	0,25
Running Costs	1	0,13	0,38	0,25	0,13	0,00
Infrastructure	1	0,40	0,13	0,27	0,40	0,00
Enterprise Culture	1	0,50	0,17	0,33	0,17	0,33
Technology	1	0,54	0,18	0,36	0,18	0,36
Geographic Interaction	1	0,10	0,50	0,40	0,30	0,20
IT Strategy	1	0,58	0,12	0,46	0,19	0,39
Business Strategy	1	0,54	0,18	0,36	0,18	0,36
Team Experience	1	0,40	0,13	0,33	0,40	0,00
Domain Knowledge	1	0,54	0,18	0,36	0,18	0,36
E-Process knowledge	1	0,19	0,03	0,16	0,58	0,39
Development Time	1	0,60	0,12	0,48	0,20	0,40
Functionality	1	0,60	0,12	0,48	0,20	0,40
Manageability	1	0,60	0,12	0,48	0,20	0,40
Quality assurance	1	0,55	0,18	0,37	0,18	0,38
Adjustability	1	0,60	0,12	0,48	0,20	0,40
Exchangeability	1	0,38	0,13	0,25	0,13	0,25
Map ability	1	0,10	0,30	0,20	0,30	0,20
Stability	1	0,54	0,11	0,43	0,18	0,36
Tool support	1	0,17	0,50	0,33	0,17	0,00
Documentation	1	0,40	0,14	0,26	0,40	0,00
				9,77		7,72

From the results in table 1 we can now determine the “winning” eCIS that can be used to identify the e-Process to use for our problem case. The table shows only 2 historical cases, but the process allows for the whole Case Base to be investigated and then the winning 4 cases can be further analysed. We now take our two cases to the next step. Table 2 shows that when a detailed analysis was executed on the low level quality aspects example case 1 was identified as the winner. For simplicity a weight of 1 was used for all low level quality aspects in this example.

Decision and Conclusions

The recommendation for the problem at hand is thus the storyboarding and user profiling be used. In a real problem scenario both these results (and even more) will be provided with the feedback on how the implementation went in order to recommend a best suited e-Process. More detail about the actual problem, before using the quality aspects will most probably be used.

Conclusions and Future Work

In this paper we have shown how methods originating from different areas of computing can be integrated into a coherent method for selecting a most suitable one out of a set of admissible e-Processes. Certainly our method is still incomplete, as we currently only have applied it to a small number of cases. However, as more case studies are performed the case base will be able provide better results. The tool will also be further improved. We furthermore propose checking alternative uses for our method such as to use it for qualification planning and as input for IT strategy discussion. That might be based on routine investigations performed for all new projects and qualification needs as well as potential goals identified. Our method is also interesting since it is a contribution to a formal model pragmatics.

References

Aamodt, A. and Plaza, E. Case-based reasoning: foundational issues, methodological variations, and system approaches. AICom – Artificial Intelligence Communications, vol. 7, 1. pp. 39 – 59.

Al-Humaidan, F. and B. N. Rossiter (2000). "A taxonomy and evaluation for system analysis methodologies in a workflow context: SSADM, UML,SSM,OPM." www.cs.ncl.ac.uk/research/pubs/trs/papers/751.pdf.

Albertyn, F. (2005). Ontology for the selection of e-Processes. Web Information Systems Engineering - WISE 2005 Workshops, New York, G. Y. Dean M., Jun W., Kaschek R., Krishnawasmy S., Pan Z., Sheng M.

Albertyn, F. and R. Kaschek (2005). eCIS development: refining e-Process selection. IADIS International Conference - e-Society 2005, Qawra, Malta,, P. K. a. M. McPherson 972-8939-03-5.

Albertyn F. (2007) E-Process Selection using Case-Based Reasoning Techniques. *IADIS international conference E-Society 2007.* Lisbon, Portugal. 3 to 6 July 2007. (http://www.esociety-conf.org/) part of the IADIS Multi Conference on Computer Science and Information Systems (MCCSIS 2007) Lisbon, Portugal, 3 - 8 July 2007 (http://www.mccsis.org) .

Albertyn, F. and R. Kaschek (2004). Ontology based e-Process selection. The Sixth International Conference on Information Integration and Web-based Applications & Services (iiWAS2004), Jakarta, Indonesia, G. Kotsis, S. Bressan, D. Taniar and I. K. Ibrahim3-85403-183-01.

Albertyn, F. and S. Zlatkin (2004). The process of developing a business processes assembler. ISTA 2004: 3rd International Conference on Information Systems Technology and Its Applications., Salt Lake City, Utah, USA July 15-17, 2004.

Antón, A. I., R. A. Carter, et al. (2001). EPRAM: evolutionary prototyping risk analysis & mitigation

(e-Commerce Software Development Process Document)

Software Requirements Engineering Lab, North Carolina State University, Department of Computer Science. Technical Report #TR-2001-08

Armengol, E., S. Onta˜non, et al. (2004). Explaining similarity in CBR ECCBR'04, Madrid, Spain

Brynjolfsson, E. (2003). The IT productivity gap. Optimize. July 2003.

Coyle, L., D. Doyle, et al. (2004). Representing similarity for CBR in XML. Advances in case-based reasoning ECCBR 2004 : European conference on case-based reasoning No7, Madrid, ESPAGNEISBN 3-540-22882-9

Cunningham, P. and A. Bonzano (1999). Knowledge engineering in a real world casebased reasoning application. Technical Report TCD-CS-1999-36, Trinity College, Dublin, 1999.

Diaz-Agudo, B. and P. A. Gonzalez-Calero (2000). "An architecture for knowledge intensive CBR systems." www.idi.ntnu.no/~agnar/tdt55/2000_ewcbr_belen-2.pdf.

Dumke, R. R. and E. Foltin (1996). "Metrics-based evaluation of object-oriented software development methods." ivs.cs.uni-magdemurg.de/sw-eng/agruppe/forshung/paper/ooevel.pdf.

Ghezzi, C., Jazyeri, M., et al.: Software qualities and principles. In: Tucker, A.B.: Computer Science Handbook. Chapman & Hall/CRC (2004)

Gruhn, V. and L. Schope (2002). "Software processes for the development of electronic commerce systems." Information and Software Technology 44(2002): 891 - 901.

Harrison, I. (1997). "Case based reasoning." www.aiai.ed.ac.uk/links/cbr.htm/#intro.Inductive Solutions, I. (2006). "

Induce-It - Case based reasoning FAQ " http://www.inductive.com/softcase.htm.

Kaschek, R., Albertyn, F., Shekhovtsov, V., Zlatkin, S. An e-Process selection model. Manuscript, Klagenfurt (2007).

Kaschek, R. and H. C. Mayr (1996). A characterization of OOA tools. Proceedings of the Fourth International Symposium on Assessment of Software Tools (SAST '96)

Kaschek, R. and H. C. Mayr (1998). "Characteristics of object oriented modeling methods." EMISA Forum 1998(2): 10-39.

Kaschek, R., Pavlov, R., Shekhovtsov, V., Zlatkin, S.: Towards selecting among business process modeling methodologies. 9th International Conference on Business Information Systems (BIS 2006), Klagenfurt, Austria (2006).

Kaschek R., Pavlov R., Shekhovtsov V. A., Zlatkin S.: Characterization and tool supported selection of business process modeling methodologies. In Witold Abramowicz, Heinrich Mayr (eds.). *Technologies for business information systems*. Springer Verlag, (2007).

Leake, D. B. (1996a). CBR in context: the present and future. Case based reasoning experiences - lessons and future experiences. D. Leake. Cambridge, MIT Press.

Leake, D. B. (1996b). "What case-based reasoning really means: now and the future " http://www.cis.ufl.edu/~ddd/FLAIRS/FLAIRS-96/leake.htm.

Loon, H. (2004).”Process Assessment and Iso/Iec 15504”. Berlin: Springer ISBN 0387231722.

Macaya, D., J. Melendez, et al. (2002). Case based approach for supervision. Application to PID controllers. 15th-IFAC World Congress, Barcelona

MacCormack, A. M., C. F. Kemerer, et al. (2003). "Trade-offs between productivity and quality in selecting software development practices." IEEE Software September/October 2003: p. 78 - 85.

Mansar, S. L., F. Marir, et al. (2003). "Case-based reasoning as a technique for knowledge management in business process redesign." Academic Conferences Limited 2003 www.ejkm.com.

Mansar, S. L., H. A. Reijers, et al. (2005). BPR implementation: a decision-making strategy. Proceedings of the Workshop on Business Process Design 2005

Mattila, S. (1998). Secure electronic transaction terms. http://www.its.bldrdoc.gov/projects/devglossary/_e-commerce.html.

Richter, M. M. (1998). Introduction - the basic concepts of CBR. Case-based reasoning technology: From foundations to applications. M. Bartsch-Sporl and S. Wess, Springer-Verlag. LNCS vol. 1400.

Saaty, T. L. (1990). *Multicriteria Decision Making: The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation.* . Pittsburgh: RWS Publications.

Watson, I. and F. Marir (1994). Case-based reasoning: a review. " http://www.surveying.salford.ac.uk/ai-cbr-mirror/classroom/cbr-review.html.

Wikipedia (2007). http://en.wikipedia.org/wiki/ISO_15504

Copyright