Cots Products To Trace Method Enactment: Review And Selection

Ghazaleh Khodabandelou, Charlotte Hug, Rebecca Deneckere, Camille Salinesi, Marko Bajec, Elena Kornyshova, Marko Janković

COTS PRODUCTS TO TRACE METHOD ENACTMENT: REVIEW AND SELECTION

Khodabandelou Ghazaleh, Centre de Recherche en Informatique, Université Paris 1 Panthéon-Sorbonne, 90 rue de Tolbiac, 75013 Paris, France, Ghazaleh.Khodabandelou@malix.univ-paris1.fr

Hug Charlotte, Centre de Recherche en Informatique, Université Paris 1 Panthéon-Sorbonne, 90 rue de Tolbiac, 75013 Paris, France, Charlotte.Hug@univ-paris1.fr

Deneckère Rebecca, Centre de Recherche en Informatique, Université Paris 1 Panthéon-Sorbonne, 90 rue de Tolbiac, 75013 Paris, France, Deneckère.Rebecca@univ-paris1.fr

Salinesi Camille, Centre de Recherche en Informatique, Université Paris 1 Panthéon-Sorbonne, 90 rue de Tolbiac, 75013 Paris, France, Camille.Salinesi@univ-paris1.fr

Bajec Marko, Laboratory for Data Technologies, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia, marko.bajec@fri.uni-lj.si

Kornyshova Elena, Centre de Recherche en Informatique, Université Paris 1 Panthéon-Sorbonne, 90 rue de Tolbiac, 75013 Paris, France, Elena.Kornyshova@univ-paris1.fr

Janković Marko, Laboratory for Data Technologies, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia, Marko.Janković@fri.uni-lj.si

Abstract

Observing information systems projects shows that Information Systems Engineering (ISE) methods are underused. The iAMF project aims at (a) tracing stakeholders' activities to identify whether this statement is true and (b) proposing more efficient ISE methods. To trace stakeholders' activities, we need a tool able to record any computerized actions - as opening applications, modifying documents, compiling programs, etc. This paper presents a review of trace-based tools that was undertaken to address the issue of recording information systems engineering methods enactment. We followed the MADISE decision making approach to select the most appropriate trace-based tool for the iAMF project.

Keywords: trace based tool, method engineering, tool review, decision-making

1 Introduction

Analysing activities in the Information Technologies (IT) context is a very lucrative business. While surfing on websites, all our activities - products and services searches, on-line game activities, comments and posts on blogs or social networks, among others - are traced and analysed to propose us the most adapted advertisements (Büchner and Mulvenna, 1998).

The approach of tracing and analysing activities has been transposed to other domains, such as software development. For instance, Weijters and van der Aalst (2003) have proposed methods and tools to analyse event logs in order to discover process models. This approach is mostly based on Petri Nets model. Whereas these are probably adequate for well-structured administrative processes, we believe this is too restrictive for engineering processes, as it reduces them to sequences of activities whereas they are by nature highly creative and decisional (Grosz et al. 1997). Tracing engineering methods enactment is a different issue, and calls for different methods, techniques and tools than tracing business processes activities (Gehlert et al. 2009). Indeed, engineers have different ways of dealing with issues that they will use depending on the situation (combined or in isolation, as such or adapted, etc).

The goal of iAMF project is to propose process mining algorithms that analyse activity traces and formalize the intentions behind them. In order to achieve this, we need (a) a tool that traces the engineering activities during software or information systems projects, and (b) a way to reconstruct intentional processes (process models specified with the intentional paradigm) out of traces recorded by the tools. We call the overall activity *intention mining*.

First, in order to be able to trace users'activities, we need to select a tool which will help us to record the appropriate activities. The MADISE methodology (Kornyshova, 2011) embeds a complete decision making process: the alternatives and the criteria selection, the evaluation of the alternatives confronted with the criteria and the final decision-making. In this paper, we show how we have followed a path in the MADISE process to select a tool to trace method enactment within the iAMF project.

This paper mainly focuses on the selection of trace-based tools. The next section presents the context of the iAMF project. Then, section 3 describes the methodology that was used to evaluate trace-based tools for ISE execution, and the result of our evaluation. The related works and ethical problems are reported in section 4. The conclusion section discusses future works.

2 Context: the iAMF project

For decades, method engineering researches have focused on providing new methods for Information System Engineering (ISE) development to improve the quality of IS products and to obtain better results in IT projects (Brinkkemper, 1996) (Jaufman et al. 2004) (Rolland, 1998) (Jarke et al. 1994) (Armenise et al. 1993). Later on, researches have addressed the issue of specifying methods in a way that would make them more flexible and more easily adapted to projects situations (Ralyté et al. 2003) (Punter and Lemmen, 1996). The real motivations to develop methods adapted to project situations are both the need of a better productivity, and the production of better quality IS. This field is known as Situational Method Engineering (SME). It helps developing proper methods suited to the environment, characteristics of projects and requirements of organizations. It assembles (e.g. by composition, refinement, or other techniques) reusable method fragments, chunks, components or services, stored in method bases (Saeki et al. 1993). Brinkkemper's definition of method engineering is "[…] the engineering discipline to design, construct and adapt methods, techniques and tools for the development of information systems." (Ralyté, 2001), (Negoro, 2001) (Brinkkemper et al. 1998), (Harmsen, 1997), (Punter and Lemmen, 1996), (Saeki et al. 1993) (Firesmith and Henderson-Sellers, 2002) (Deneckère et al. 2008) proposed and developed a number of approaches of SME.

Despite the fact that these methods are designed and elaborated to be adapted, empirical researches show their application in practice is quite rare (Mirbel and de Rivieres, 2002) (Ralyté and Rolland, 2001). Nowadays, theoretical methods in software development projects are seldom applied or followed systematically and we do not have any idea about their adaptation regarding the projects and organizations context. Consequently, we do not know which methods are used. In fact, we do not even have systematic formal proof that they properly address the issues met in projects. This has caused a number of failures in IT projects and, more importantly, it is a catalyst to low quality IT products (Standish Group, 2011). Nevertheless, applying SME approaches taking into account the specificities of a given organization is time-consuming because it requires a considerable commitment of the stakeholders (developers, IT managers…) due to their difficult deployment. The practitioners or stakeholders must understand all the concepts of a method as a whole even if they use only one fragment (Mirbel and de Rivieres, 2002).

As a response to ill-used SME in practice, researchers at the University of Ljubljana developed an innovative approach, called Agile Method Framework (AMF) (Bajec et al. 2007). The vision of this approach was to bring SME principles closer to practitioners, by considering social and technical aspects that influence how practitioners perceive usefulness of methods as guidelines for their everyday work. After few years of the application of AMF in practice, the results showed that even perceived as useful, AMF was not fully embraced in the participating software companies. One of the reasons was that AMF still required active involvement of developers, which consumed their limited time for the software development.

The new iAMF project (intelligent AMF), which we designed together with the colleagues from University of Ljubljana, starts from the above findings. Its main goal is to make the integration of SME techniques into software development life cycle more transparent for the developers, so that they would not perceive them as a burden, but rather as a way to support their everyday activities. Consequently, this would lead to higher quality of software and lower the risk for project failures.

The main idea of the iAMF is to learn from the observation of developers and other stakeholders, what they really do on projects, how they act and perform, and based on this observation to automatically create a formalized method that will best cover their needs for future projects. Our expectation is that if prescribed methods are created this way, the developers will follow them much more rigorously, which again, will raise the chance for the project success. Figure 1 illustrates the iAMF approach. Trace-based tools allow recording the traces of ISE project activities of different stakeholders in a traces base. By applying intention mining algorithms and classification methods on these traces, we can discover the real methods used in a given project.

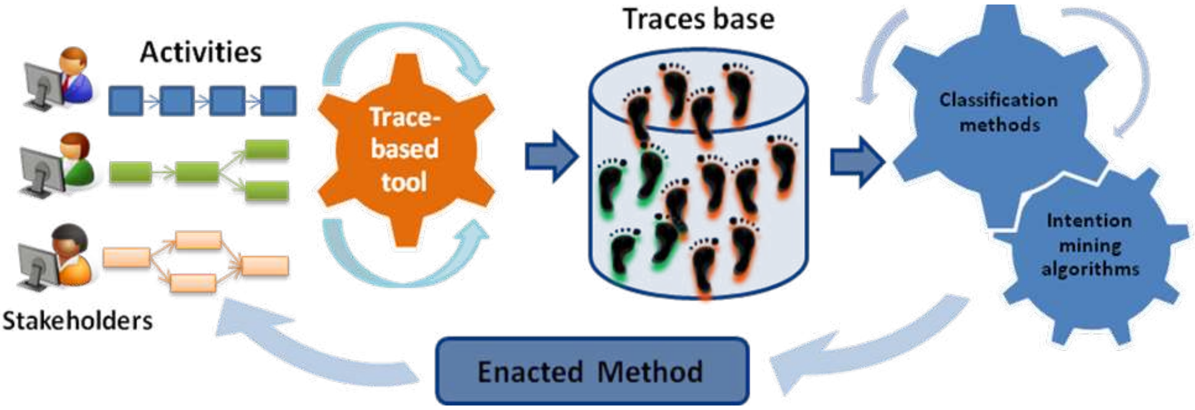

Figure 1. Representation of iAMF approach.

The goal of this paper, which is a first brick of iAMF project, consists in selecting the appropriate tool to record the traces of stakeholders' activities while enacting information systems engineering methods. In order to select the appropriate tool, we follow the MADISE decision-making method.

3 Decision Making Methodology

Our aim is to collect data from user activities that correspond to any event produced by a tool. The data thereby collected should be analyzed to extract useful information about the method used. To retrieve the event logs, we need the appropriate tool. For lack of human and materiel resources, we cannot develop a tool from scratch; instead we will select a tool among the existing ones. We are then looking for Commercial Off-The-Shelf (COTS) tools able to record the enactment of users' activities. Reusing components of COTS software is profitable in terms of cost and time. Compared with custom-made software, COTS software provides standardized functionalities, enhanced responsiveness and thus, an important gain of time (Kesseler, 2008). However, matching COTS products to stakeholders' requirements is a complex task.

We then have to establish a list of criteria needed to identify the tools that match customers' requirements, that is to say the requirements of the iAMF project. At this stage, we face a component selection problem than we can address as a Multi Criteria Decision Making problem. Several approaches have been proposed, to deal with this. For instance, Kontio (1995) proposed the OTSO (On-The-Shelf-Option) method but there is a lack of well-defined process in order to provide requirements acquisition and non-functional requirements (Alves and Castro 2001). STACE (Social-Technical Approach to COTS Evaluation) framework, proposed by Kunda and Brooks (1999), has the same problem and it does not utilize a decision-making technique to evaluate products. To select a COTS product taking into account the criteria, (Maiden and Ncube 1998) proposed the PORE method (Procurement-Oriented Requirements Engineering). It provides some techniques for requirements acquisition but this method does not support a good description of non-functional requirements. We finally choose to follow the MADISE methodology (Kornyshova et al. 2011, Kornyshova 2011). MADISE embeds a complete decision process, adaptable to our context and quite easy to follow. Moreover, its calculation complexity is lower than others methods such as Analytic Hierarchy Process (AHP) (Satty, 1990) - using AHP would have lead us to compare each pair of tools for each requirement, leading to a high number of comparisons (308 comparisons for 8 tools and 11 requirements in our case $((8^2 - 8)/2)*11$).

Figure 2 shows the path followed in MADISE to prescribe the most appropriate tool to meet the iAMF project requirements.

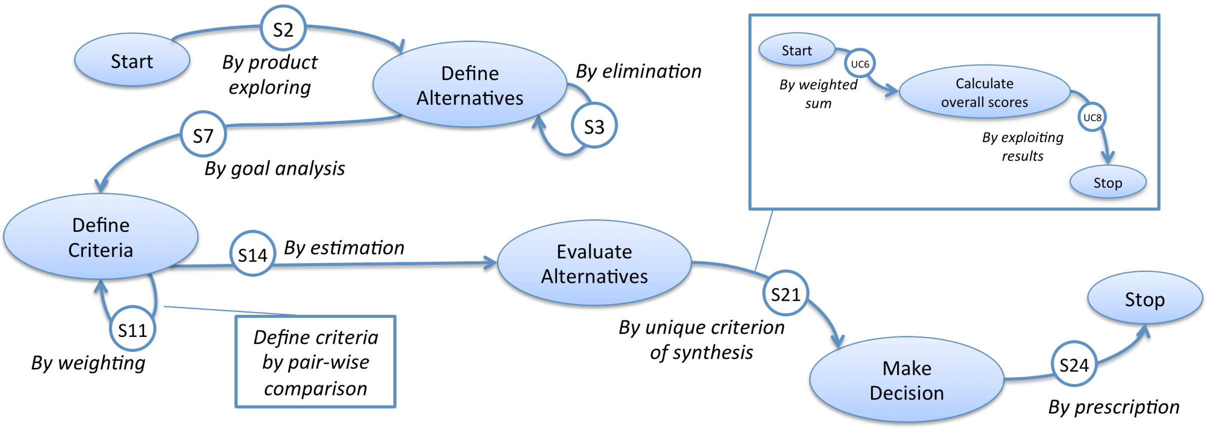

Figure 2. Extract of MADISE (Kornyshova, 2011) used to select and evaluate trace-based tools.

First, we defined alternatives by exploring the different trace-based tools available (the alternatives of our decision-making problem) (S2). Second, we eliminated some of them because they were not adapted at all to our needs (S3). According to our requirements (S7), we defined 15 criteria (in our case, requirements addressed as goals) to evaluate the potential tools. We used pair-wise comparisons to weight the different criteria previously found (S11). Then, we had to evaluate the alternatives (the tools) according to the criteria (the requirements) by estimating the conformance of each tool with each criterion (S14). We calculated the overall scores of each alternative in order to select and prescribe the most adequate tool (S21, S24). The S21 section, which allows evaluating the alternatives, is refined in another map (cf. blue rectangle) to calculate the overall scores of each tool to select the best suited one. These steps are detailed in the next sections.

3.1 Define alternatives: Tools identification

This step corresponds to the execution of the MADISE components S2 (Define alternatives list by product exploring) and S3 (Refine alternative list by elimination).

In modern organizations IT infrastructure, logs are one of the crucial axes to provide vital information about process model-in-action as they provide an insight about the system operations and users behaviours. It is obvious that the task of recording these logs requires an automated tool. In order to identify existing event logs analysers to define the first set of potential alternatives, we conducted researches in different articles, references and on the Web. The keywords were "event logs analysis", "event logs analyser", "free event logs analyser", "event logs COTS tool", "workflow tool", "trace-based tool" and "trace-based system framework". We found different types of tools: web log analysers, workflow engines and security-based tools. We also found some academic frameworks, such as ProM (van Dongen et al. 2005) and kTBS[1]. Our Web searches also pointed out to web analytics tools.

3.1.1 Define alternatives list by product exploring

Web log analysers or web log analysis software parse server log files from web servers, extract and report information to network analysts. This information corresponds to indicators about number of visits on webservers, duration of visits, hosts list, operating systems, browsers and many other details about operations done by users on web server. These kinds of analysers do not meet our needs - our goal is not a web-oriented monitoring of users activities - so we definitely exclude them from our research field.

Another kind of event logs analysers is workflow engines that allow designing models, managing, following, analysing and monitoring the business processes of companies in a computerized way. Their objective is to automate business processes according to predefined procedures. They create models to assign given tasks to given project members, to manage users, to send reports and to make statistics. A workflow engine requires a pre-modelled process - a process must be modelled before being implemented in a workflow engine. The reasons why we do not focus our research in workflow engines are justified by their restriction to the predefined models, their non-suitability for creative works (but repetitive tasks) as they are based on sequence of activities, the generation of high-level event logs, their rigidity and their linearity - there is no place for users' intention. Among the existing workflow-based tools, Business Activity Monitoring and Business Process Intelligence are the most used. Many others have emerged, such as Process Miner, Little Thumb, EMiT, InWoLvE and MinSoN, which are academic tools. Among commercial tools we can cite ARIS PPM, HP BPI, and ILOG JViews.

kTBS[1] framework is a kind of Data Base Management System able to record threads and process from kernel of systems. The proposed method consists in defining a model for the traces. This model defines type of observed elements (obsel), associated attributes and possible relationships between the elements. In this method, a transformation phase should be carried out on traces. It is used for audio and visual recordings from users of specifically designed systems (e-learning systems for example) in order to provide recommendations. This approach is qualitative in the sense that the analysis of the material cannot be processed by a computer. It is thus very time consuming, and does not meet our needs as we do not want to record any audio or visual material.

3.1.2 Refine alternatives list by elimination

Security-based tools are designed to ensure the security of IS. They propose a number of different technologies such as detection of intrusion, file integrity checking, policy monitoring, real-time

[1] kTBS, http://kernel-for-trace-based-systems.readthedocs.org/en/latest/index.html, consulted November 2012.

alerting, secure transmission, database support, security of sensitive files, identification of events, etc. Security-based tools generate low-level event logs and are not limited to any predefined models. These tools are very flexible as they can record any kind of event. We propose to use them to record and extract information about stakeholders' activities development during ISE projects.

Within the large amount of log analysers, we carried out empirical researches to analyse their features and their operating mode (for both open source and commercial tools) in several websites, as (Chuvakin, 2011). Many tools are enhanced versions of ancient ones, as rsyslog[2] is the improvement of Syslog that uses traditional format for syslog.conf configuration files. Subsequently, rsyslog is an associated front-end of LogAnalyzer[3]. Some of these tools are designed for mining frequent patterns such as LogHound[4] or SLCT[5]. Among some ancient tools we quote Logwatch[6], Lire[7] and LogSurfer[8].

Hereafter, we introduce a brief description of some of the log analyser tools that seem to be the most adapted to our requirements.

Ossec[9] is a tool to analyse logs for indications of possible security breaches. It is not a log management tool but it stores some alerts and perform analysis of real-time data log from UNIX systems, Windows servers and network devices. It is an open source host-based intrusion detection system that performs rootkit detection, real-time alerting and active response. Despite the variety of functionalities of Ossec, this tool is neither able to identify from which application events are produced from nor capture modifications on the user system. It just shows when an action (open/close) was executed and by whom.

Snare[10] (System iNtrusion Analysis and Reporting Environment) uses a set of agents to collect event logs at real-time to provide Security Information and Event Management. It facilitates centralizing data logs. It collects and filters events logs and forwards them to a central server. Agents of Snare are available for Linux, Windows, Solaris and IIS, among others.

Syslog-ng[11] is an infrastructure for log management. It securely offers to collect, filter, classify, store and forward log messages via IS environments. Syslog-ng can deal with correlating problem by using classification and message parsing.

EventLog analyzer[12] provides System Information and Event Management (SIEM). This software collects, analyses, searches, and reports event logs generated by IS machines. It affords intelligent monitoring and produces reports such as user activity reports, regularity compliance reports, historical trend reports and so on.

3.2 Define criteria: Requirements identification and weighting

This step corresponds to the execution of two components of MADISE: S7 (Define criteria by goal analysis) and S11 (Define criteria by weighting). This last component is tactical and is refined in our case with a component allowing defining criteria by pair-wise comparison.

3.2.1 Define criteria by goal analysis

Despite endeavour of analysts to formalize a system that meets user requirements, there is often a mismatch between user actual requirements and the system-in-action. This is due to the lack of a

[2] rsyslog, http://www.rsyslog.com/, event logs analyzer, viewed October 2012.
[3] Log Analyzer, http://freecode.com/projects/phplogcon, event logs analyzer, viewed October 2012.
[4] LogHound, http://ristov.users.sourceforge.net/loghound/, event log analyzer, viewed October 2012.
[5] SLCT, http://ristov.users.sourceforge.net/slct/, event log analyzer, viewed October 2012.
[6] Logwatch, http://sourceforge.net/projects/logwatch/, event log analyzer, viewed October 2012.
[7] LogReport, http://logreport.org/, event log analyzer, viewed October 2012.
[8] Logsurger, http://www.crypt.gen.nz/logsurfer/, event log analyzer, viewed October 2012.
[9] Ossec, http://www.ossec.net/, event log analyzer, viewed October 2012
[10] Snare, http://intersectalliance.com/projects/index.html, event log analyzer, viewed October 2012.
[11] Syslog-ng, balabit.com/network-security/syslog-ng/, viewed October 2012.
[12] EventLog analyzer, http://www.manageengine.com/products/eventlog/, viewed October 2012.

rigorous requirements analysis before launching the design phase of the projects. A requirement, according to Herrmann and Daneva (2008), is characterized by several properties considered for requirements prioritization (Karlsson and Ryan, 1997): type of requirements (e.g. functional/non-functional or primary/secondary requirements), assessment of benefit for stakeholders, assessment of software size that embeds the requirement, assessment of cost to build what embeds the requirement, priority, and requirement dependencies (i.e. the dependency between requirements is the degree of satisfaction when one requirement influences the cost caused or the benefit added by another one).

To avoid manual data collection (risk of confusion), to have a global view on data (for further comparison) and to monitor the collection of traces more easily, we need a centralized data collection infrastructure (e.g. central servers, routers, switches, applications…). This infrastructure should allow a remote access to collect event logs from client side and network devices, applications and OS. Hence one of the requirements is client/server architecture. UNIX-based servers usually support all these functionalities but we keep Windows-based servers as users tend to use Windows to carry out their tasks (OS platform statics, 2012).

The data communication between client and central server is established through a protocol of transport. Transport protocols such as User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) are the most commonly used in network connections. TCP provides data integrity and guarantee the data delivery by control flow. UDP is faster than TCP because there is neither control flow or correction error, thus UDP uses a network bandwidth shorter than TCP.

To record IS engineering method enactment, we need to monitor users' activities; it is thus important to know which applications are used. Such data brings the important information not only about users' activities but also about users' intentions. Stakeholders execute specific actions due to their objectives and prescribed methods using a number of different applications. Generally, we consider that in IT projects, stakeholders use two categories of applications in order to execute their assigned tasks: ISE applications are used to produce the Information System itself such as Project Management Tools (MsProject, SVN, Microsoft SharePoint…), Modelling Tools (Visual paradigm, IBM rational, Mega…), Integrated Development Environment (NetBeans, Eclipse…), or even Computer-Aided Methods Engineering tools, (MetaEdit+, Eclipse Process Framework…); non ISE applications support the ISE process but are not directly related to the production of the Information System such as Office Tools, Mail, Chat, Browser, Intranet (Lotus…), Knowledge Management System. The stakeholders manipulate these applications through many actions. In other words, the actions carried out by users within the applications lead us to understand their activities and intentions in a given context. These actions could directly involve the operating system (e.g. open or close a given application) or be limited to internal actions (e.g. save or modify a document, open a project within a tool…). The possible requirements are defined below, each "Ri" representing a requirement:

Functional requirements	Application	The trace-based tool should be able to report events when users install/ uninstall/ reinstall an application (R1), update an application (R2) and manipulate an application (actions as open/close/save/modify) (R3).
Non-functional requirements	OS Kernel (client side):	The trace-based tool should be multi-platform (UNIX: Linux/AIX/Solaris/BSD/MC OS, Windows) (R4), free/open source (R5) and collect event logs from system error (R6).
	Hardware	The tool should be able to report logs when a device is installed/ uninstalled on the users machines (R7) and send a message error (missing peripheral) (R8).
	Network	The tool should be able to report logs when security event happens (firewall, Intrusion Detection System, antivirus, proxy servers, Web servers, Syslog) (R9), report logs when browser event happens (visiting websites, checking mail) (R10), support secured communication (transport protocols TCP, ping, trace root, SNMP) (R11) and provide a web-based human interface (R12).
	Server	Server agent of tool should be multi-platform (UNIX: Linux/AIX/Solaris/BSD/MC OS, Windows) (R13), free/open source (R14) and should have a database to store the event logs (R15).

3.2.2 Define criteria by weighting

The next step consists in constructing a "pair-wise" comparison matrix from criteria list to assess their relative value. As we are leading the iAMF project, we have the expertise to define the weight between the requirements. We assign an arbitrary weight between 1 and 9 to each {Ri, Rj} pair of criteria. This weight represents relative importance, in other words, how much Ri is preferred to Rj or reciprocal. Thereby a normalized set of weights is established to fill the evaluation matrix. The importance of Ri over Rj is graded from 1 to 9: 1 – equal importance (the contribution degree of Ri and Rj is equal); 3 – slightly higher (Ri is slightly promoted over Rj); 5 – strongly higher (Ri is strongly promoted over Rj); 7 – very strongly higher (Ri is very strongly promoted over Rj); 9 – absolutely higher (Ri is absolutely promoted over Rj). The values 2, 4, 6 and 8 are intermediates scales considered as trade-off.

Table 1 presents the results of the definition of the criteria weighting by pair-wise comparison. It should be read as “R1 requirement is sixth times more important than R2 requirement” (in grey in Table 1). The opposite is “R2 requirement is sixth times less important than R1 requirement”.

Rj / *Ri*	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**	**R15**	**Sci**
R1	1	6	1/8	1/3	1/3	9	7	9	8	1/5	1/4	3	1/3	3	3	.05974
R2	1/6	1	1/9	1/8	1/7	9	6	9	9	1/7	1/8	1/3	1/7	1/3	1/5	.0329
R3	8	9	1	6	7	9	9	9	9	7	3	2	3	7	6	.2218
R4	3	8	1/6	1	3	6	8	9	9	3	2	1/3	1	3	4	.1012
R5	3	7	1/7	1/3	1	9	7	9	9	1/3	3	6	1	3	4	.0976
R6	1/9	1/9	1/9	1/6	1/9	1	1	1	1	1/9	1/7	1/7	1/9	1/5	1/5	.0095
R7	1/7	1/6	1/9	1/8	1/7	1	1	1	1	1/7	1/8	1/8	1/9	1/6	1/6	.0095
R8	1/9	1/9	1/9	1/9	1/9	1	1	1	1	1/8	1/8	1/8	1/9	1/5	1/5	.0092
R9	1/8	1/9	1/9	1/9	1/9	1	1	1	1	1/8	1/8	1/8	1/9	1/5	1/5	.0092
R10	5	7	1/7	1/3	3	9	7	9	8	1	1	5	3	6	6	.1186
R11	4	8	1/3	1/2	1/3	7	8	9	8	1	1	4	1	5	5	.0932
R12	1/3	3	1/2	3	1/6	7	8	9	8	1/5	1/4	1	1/3	5	5	.0746
R13	3	7	1/3	1	1	9	9	9	9	1/3	1	3	1	6	6	.0953
R14	1/3	3	1/7	1/3	1/3	5	6	5	5	1/6	1/5	1/5	1/6	1	1/3	.0305
R15	1/3	5	1/6	1/4	1/4	5	6	5	5	1/6	1/5	1/5	1/6	3	1	.0364

Table 1. Value matrix.

All the values of the matrix should be averaged over normalized columns (Figure 3a). Then by averaging the rows (i.e. dividing the row sum by the total number of requirements - 15), we obtain the scores (Si) of each requirement nRi (Figure 3b).

$$nRi = Rj * 1/\sum_{j=1}^{15} Rj \qquad\qquad Sci = \frac{1}{15} * \sum_{i=1}^{15} nRi$$

Figure 3.a) Normalization equation. b) Calculate the total score equation.

3.3 Evaluate alternatives: Tools evaluation

This step corresponds to the execution of the component S14 of MADISE (Evaluate alternatives by estimation). At this stage, the alternatives (tools argued in Section 3.1) are evaluated according to their contribution to the satisfaction of the requirements' (the weighted criteria obtained in Section 3.2) using heuristics, i.e. based on the human judgment. We have selected to use evaluation scores (Ei) going from 0 to 1: tools may provide any satisfaction (0), low satisfaction (0.25), medium satisfaction (0.50), high satisfaction (0.75), and very high satisfaction (1). Table 2 shows the estimation value assigned to each tool for each requirement.

3.4 Make decision: Overall score evaluation

This step represents the execution of the S21 MADISE component (Make decision by unique criterion of synthesis). In our case, this component leads to the execution of the component "calculate overall scores by weighting sum".

Table 2 shows the total evaluation score (TEi) of each tool regarding the weighted requirements obtained in section 3.2 (Sci) and using the evaluation scores defined in section 3.3 (Ei). We do not take into account R6, R7, R8 and R9 because their related scores are lower than 0.01 (see table 1) and that means they are not enough important to be considered as criteria. We calculated the score of each tool (TEi) as a weighted sum of the evaluation score (Sci) reflecting the level of satisfaction that has been provided by each tool.

	Weight (Sci)	**Ossec**	**Snare**	**Syslog-ng**	**Advanced Event Viewer**	**EventLog analyzer**	**Event Log Monitor**	**GFI event manager**	**TNTSoft ware**
R1	0.05974	0*Sc1	1*Sc1	0.5*Sc1	0.75*Sc1	0.75*Sc1	0.75*Sc1	0.5*Sc1	0.75*Sc1
R2	0.0329	0*Sc2	1*Sc2	0.5*Sc2	0.75*Sc2	0.75*Sc2	0.75*Sc2	0.5*Sc2	0.75*Sc2
R3	0.2218	0.25*Sc3	0.75*Sc3	0.25*Sc3	0.25*Sc3	1*Sc3	0.5*Sc3	0.75*Sc3	0.5*Sc3
R4	0.1012	1*Sc4	1*Sc4	1*Sc4	0.5*Sc4	1*Sc4	0.5*Sc4	0.25*Sc4	0.5*Sc4
R5	0.0976	1*Sc5	0,75*Sc5	0.5*Sc5	0.25*Sc5	0.25*Sc5	0*Sc5	0*Sc5	0*Sc5
R10	0.1186	1*Sc10	1*Sc10	0.5*Sc10	0.25*Sc10	0.5*Sc10	0.75*Sc10	1*Sc10	1*Sc10
R11	0.0932	1*Sc11	1*Sc11	1*Sc11	1*Sc11	1*Sc11	1*Sc11	1*Sc11	1*Sc11
R12	0.0746	1*Sc12	1*Sc12	1*Sc12	0.25*Sc12	1*Sc12	1*Sc12	0.5*Sc12	1*Sc12
R13	0.0953	1*Sc13	1*Sc13	1*Sc13	0.5*Sc13	1*Sc13	0.5*Sc13	0.5*Sc13	0.5*Sc13
R14	0.0305	1*Sc14	1*Sc14	0.5*Sc14	0.25*Sc14	0.25*Sc14	0.5*Sc14	0*Sc14	0*Sc14
R15	0.0364	1*Sc15	0,75*Sc15	1*Sc15	0.75*Sc15	1*Sc15	1*Sc15	1*Sc15	1*Sc15
TSi		0.647	**0.874**	0.626	0.424	0.783	0.572	0.571	0.601

Table 2. Tools comparison.

3.5 Result analysis and tool prescription

This step represents the execution of the S24 MADISE component (Prescribe decision by validation). Studying the tools comparison table (Table 2), we can identify the tool that seems the most appropriate for the iAMF project. It score is 0.874 that means that Snare fits at 87,4 %; as a result, we prescribe this tool for the project next steps. However, Snare does lack some functionality. Snare in free version for Windows is not able to audit file or folder updates (e.g. change, delete or save a document). To do so, we configure "Windows auditing for object access" in "Local Security Policy" then enable auditing in folders where we need to trace a document update. Thereby we will be able to find the information in Microsoft event viewer. In this manner, the given tool is validated by stakeholders and prescribed for its further usage for tracing methods.

4 Discussions

In this section, we discuss the related works of the iAMF project: process mining, map and intention mining and classification. We also discuss the ethical problems associated to trace recording.

4.1 Process mining

One of the key fields in process engineering is process mining (van der Aalst, 2011). This technique has emerged a few years ago to help analyse systems and has been extended to analyse business process based on event logs. The statement was that workflow engines record an increasing amount of information about business processes in form of event logs, but most organizations diagnose problems based on fictional models rather than using the facts induced by these logs. Process mining – using activity oriented process model-driven approaches and data mining - allows organizations to fully benefit from the information stored in their systems by checking the conformance of processes, detecting bottlenecks, and predicting execution problems. Logs are recorded within the workflow tools and are analysed by various algorithms such as the α-algorithm (Weijters and van der Aalst 2003) and statistics. These process mining techniques aim to discover process models based on event logs (Discovery) or compare existing process models with recorded event logs and to calculate differences between them (Conformance). Enhancement allows improving the existing process model to reduce the gap between real activities and prescribed activities.

Despite the fact that we use similar techniques in our proposed method, there is a huge difference between them: process mining is limited to sequences of activities whereas iAMF takes into consideration the intentional point of view of a process as it is based on the Map process metamodel.

4.2 Map and Intention mining

The usual point of view of IS processes is to execute linear activities. However, stakeholders have a myriad of possibilities when enacting a process and may choose to enact actions differently according to their context. In fact, they choose to enact one action or another according to their intentions. The concept of intention has been integrated in a specific kind of process models, called intentional or strategy oriented process models (as opposed to activity oriented process models). Map is an intentional process metamodel that has been introduced by Rolland et al. (1999). Map models (instances of Map metamodel) guide any actor through the enactment of any process by proposing strategies dynamically to stakeholders according to the intentions they want to achieve. As a result, the defined IS development process models have a dynamic structure; are more flexible and suitable to the adaptation of the project specifications and context of each stakeholder. In a previous paper (Hug at al., 2012), we proposed a metamodel to record intentional traces based on Map to provide recommendations using classification techniques and process mining techniques.

As mentioned above, a map process model provides some guidance but offers no recommendation. The objective of iAMF method is to provide recommendations to improve the quality of products and to help stakeholders enhancing their ways of working, based on Map process models.

4.3 Classification

The intentional aspect of process modelling plays a crucial role to obtain adapted methods. A method is adapted when it fully respects the intentions and ways of working of project stakeholders. When users are categorized according to their intentions, it is easier to detect any declination of process model regarding intention classes. However, it is too pervasive and time-consuming to collect traces and related intentions each time users carry out actions (they should precise the intentions behind those actions). Some approaches to classify traces exist; Minseok et al. (2009) proposed a method to classify event logs according to cases. The questions we raise are: what intentions can we discover about users' activities from the event logs information? How can we proceed to classify them?

Appropriate techniques to handle mentioned problems are Machine Learning (ML) techniques (Eibe and Witten 2005). Application of ML techniques on event logs are organized in two phases: training and prediction. The training phase (supervised training) consists in collecting samples (event logs) coming from users; these logs are then interpreted to identify the intentions behind them. Thereby we have a space of users' activities with related intentions; a lot of samples should be collected to ensure the accurate learning of classifier. Once the classifier is trained, we can test the classifier to ensure its performance in terms of intention classification; this is the first step of the prediction phase. To do so, we input some samples to the classifier and evaluate the result to estimate its performance. The second step of prediction phase consists in predicting the intentions of new event logs.

4.4 Ethics

Recording log traces may lead to ethical problems. Stakeholders will be reluctant to accept that their activities will be recorded so we will have to ensure that all the recorded traces are anonymous. It is then important that the protocol used to send the collected traces from the client to the server is secured. We have to ensure the data security at any time so that nobody can find who produces a trace. Our goal is not to keep under surveillance the activities of stakeholders but to understand how they work, which defined method parts are not used or which have to be improved.

5 Conclusions and perspectives

This paper is part of the iAMF project. We presented a review of trace-based tools in order to record any stakeholder's activities during ISE projects. By following the MADISE decision making approach, we were able to prescribe one tool among all the alternatives, Snare. MADISE was the most appropriate decision making method as we have been able to properly define the functional and non-functional requirements, the complexity calculations were lower than in other methods (e.g. AHP) and it covered the whole decision process. The selected tool, Snare, is intended to check security breach. We divert its aim by using it to trace users' activities in order to improve ISE methods. It still remains to define whether its diversion will be effective to help enacting users' activities.

The next phase of the iAMF project is to collect traces. We will first test the selected tool in academic environments, at the University of Ljubljana and the University of Paris 1 Panthéon-Sorbonne with students, researchers and teachers. We will then be able to test the Machine Learning and Intention Mining algorithms we are elaborating. Finally, we will install the tool in industrial environments. We will analyse the collected traces, refine the algorithms and propose new process models for ISE development projects, in order to offer stakeholders more adapted methods.

References

Alves, C. and Castro, J. (2001). CRE: a systematic method for COTS components Selection. Proc. of the XV Brazilian Symposium on Software Engineering (SBES), Rio de Janeiro.

Armenise, P., Bandinelli, S., Ghezzi, C. and Morzenti, A. (1993). A survey and assessment of software process representation formalisms. Int. Journal of Soft. Eng. and Knowledge Eng, 3(3).

Bajec, M., Vavpotič, D. and Krisper, M. (2007). Practice-driven approach for creating project-specific software development methods, Information and Software Technology, 49 (4), 345–365.

Brinkkemper, S. (1996). Method Engineering: Engineering of Information Systems Development Methods and Tools. Information and Software Technology, 38 (4), 275-280.

Brinkkemper, S., Saeki, M. and Harmsen, F. (1998). Assembly Techniques for Method Engineering. Proc. of CAiSE'98. Pisa, Italy.

Büchner, A. G. and Mulvenna, M. D. (1998). Discovering internet marketing intelligence through online analytical web usage mining. ACM Sigmod Record, 27(4), 54-61.

Chuvakin, A. (2011). Open source and free log analysis and log management tools, http://securitywarriorconsulting.com/logtools/

Deneckère, R., Iacovelli, A., Kornyshova, E. and Souveyet, C. (2008).From method fragments to method services. Proc. Of EMMSAD'08, Montpellier, France.

Eibe, F. and H.Witten, I. (2005). Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Firesmith, D.G., Henderson-Sellers, B (2002). The OPEN Process Framework. An Introduction. Addison-Wesley, London, UK, pp.330.

Gehlert, A., Schermann, M., Pohl, K. and Krcmar, H. (2009). Towards a research method for theory-driven design research. Wirtschaftsinformatik Proceedings, Paper 42.

Grosz, G., Rolland, C., Schwer, S., Souveyet, C., Plihon, V., Si-Said, S., Ben ashour, C.and Gnaho, C. (1997). Modelling the requirements engineering process: An overview of the nature approach. Req. Eng., London, 2, 115-131.

Harmsen, A.F. (1997). Situational Method Engineering, Ernst & Young Management Consultants.

Herrmann, A. and Daneva, M. (2008). Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research. 16th IEEE Int. Requirements Engineering Conf.

Hug, C., Deneckère, R. and Salinesi, C.(2012). Map-TBS: Map process enactment traces and analysis. Proc. of Sixth Int. Conf. on Research Challenges in Information Science (RCIS), 204-209, Spain.

Jarke, M., Pohl, K., Rolland, C. and Schmitt, J.R. (1994). Experienced-Based Method Evaluation and Improvement: A Process Modeling approach. Int. IFIP WG8. 1 Conf. in CRIS series: Method and associated Tools for the Information Systems Life Cycle, North Holland.

Jaufman, O., Dold, A., Haeberlein, T., Schlumpberger, C. and Stupperich, M. (2004). Requirements for flexible software development processes within large and long taking projects. QUATIC'04, Portugal.

Karlsson, J. and Ryan, K. (1997). A Cost–Value Approach for Prioritizing Requirements, Journal of IEEE software. IEEE Computer Society Press, 14 (5), 67-74, Los Alamitos, CA, USA.

Kesseler, E. (2008). Assessing COTS Software in a Certifiable Safety-Critical Domain, Information Systems Journal (18), 299-324.

Kontio, J. (1995). A COTS Selection Method and Experiences of Its Use. Proceedings of the 20th Annual Software Engineering Workshop, Maryland.

Kornyshova, E. (2011). MADISE: Method Engineering-based Approach for Enhancing Decision-Making in Information Systems Engineering, PhD thesis, University Paris I Panthéon-Sorbonne.

Kunda, D. and Brooks, L. (1999). Applying Social-Technical Approach for COTS Selection. Proceedings of the 4th UKAIS Conference, University of York.

Maiden, N.A. and Ncube, C. (1998). Acquiring COTS software selection requirements. Journal Software IEEE, 15 (2), 46-56.

Minseok, S., W. Günther, C. And van der Aalst, W.M.P. (2009). Trace Clustering in Process Mining Business Process Management Workshops, Lecture Notes in Business Inf. Processing, 17, 109-120.

Mirbel, I. and de Rivieres, V. (2002). Adapting Analysis and Design to Software Context: The Jecko Approach. In 8th Int. Conf. on Object Oriented Information Systems.

Negoro, F. (2001). Methodology to Determine Software in a Deterministic Manner. Proc. of ICH, Beijing, China.

Punter, H.T. and Lemmen, K. (1996). The MEMA model: Towards a new approach for Method engineering. Information and Software Technology, 38 (4), 295–305.

Ralyté, J. (2001). Method chunks engineering, PhD thesis, University of Paris 1 Panthéon-Sorbonne.

Ralyté, J. and Rolland, C. (2001). An Assembly Process Model for Method Engineering. Proc. of CAISE'01, Lecture Notes in Computer Science, 2068, 267-283.

Ralyté, J., Deneckère, R. and Rolland, C. (2003). Towards a Generic Model for Situational Method Engineering. Proc. of CAiSE'03, Lecture Notes in Computer Science, 2681, 95-110.

Rolland, C. (1998). A Comprehensive View of Process Engineering. Proc. of CAiSE'98, Lecture Notes in Computer Science, 1413, 1413, 1-24.

Saeki M., Iguchi, K., Wen-yin, K. and Shinohara, M. (1993). A meta-model for representing software specification and design methods. Proc. of the IFIP WG8.1 Conf. on Information Systems Development Process, 149-166.

Saaty, T.L. (1990). How to make a decision: The analytic hierarchy process. European journal of operational research, Vol.48(1),9-26.

Standish group, CHAOS Report, 2011.

van der Aalst, W.M.P (2011). Process Mining: Discovery, Conformance and Enhancement of Business Processes. 1st Edition, Springer, Berlin.

Weijters, A.J.M.M. and van der Aalst, W.M.P. (2003). Rediscovering Workflow Models from Event Based Data using Little Thumb. Integrated Computer-Aided Engineering, 10 (2), 151-162.