Evaluating Workflow Trust using Hidden Markov Modeling and Provenance Data

Mahsa Naseri and Simone A. Ludwig

Abstract In service-oriented environments, services with different functionalities are combined in a specific order to provide higher-level functionality. Keeping track of the composition process along with the data transformations and services provides a rich amount of information for later reasoning. This information, which is referred to as provenance, is of great importance and has found its way into areas of computer science such as bioinformatics, database, social, sensor networks, etc. Current exploitation and application of provenance data is limited as provenance systems have been developed mainly for specific applications. Therefore, there is a need for a multi-functional architecture, which is application-independent and can be deployed in any area. In this paper we describe the multi-functional architecture as well as one component, which we call workflow evaluation. Assessing the trust value of a workflow helps to determine its rate of reliability. Therefore, the trustworthiness of the results of a workflow will be inferred to decide whether the workflow's trust rate should be improved. The improvement can be done by replacing services with low trust levels with services with higher trust levels. We provide a new approach for evaluating workflow trust based on the Hidden Markov Model (HMM). We first present how the workflow trust evaluation can be modeled as a HMM and provide information on how the model and its associated probabilities can be assessed. Then, we investigate the behavior of our model by relaxing the stationary assumption of HMM and present another model based on non-stationary hidden Markov models. We compare the results of the two models and present our conclusions.

Mahsa Naseri
Department of Computer Science, University of Saskatchewan, Canada, e-mail: naseri@cs.usask.ca

Simone A. Ludwig
Department of Computer Science, North Dakota State University, USA, e-mail: simone.ludwig@ndsu.edu

1 Introduction

In service-oriented environments, services with different functionalities are combined in a specific order to provide higher-level functionality. The composition of services is usually referred to as workflows. A workflow is defined as the automation of the processes and involves the orchestration of a set of services, agents and actors that must be combined together to solve a problem or define a new service. Different services of the workflow represent the transformation processes that receive the data as input to produce the transformed data as output. The workflow graph often describes a network where the nodes are services and the edges represent messages or data streams that channel work or information between services. Each node processes a stream of messages and forwards the resulting streams into its connected nodes.

In such environments, great numbers of workflows are executed to perform mostly scientific and not often business experiments. The workflow activities are run repeatedly by one or more users and large numbers of result data sets in the form of data files and data parameters are produced. As the number of such datasets increases, it becomes difficult to identify and keep track of them. Besides, in these large-scale scientific computations how a result dataset is derived is of great importance as it specifies the amount of reliability that can be placed on the results. Thus, information on data collection, data usage and computational outcome of these workflows provide a rich source of information.

Capturing the execution details of these transformations is a significant advantage for using workflows. The execution details of a workflow, referred to as provenance information, is usually traced automatically and stored in provenance stores. Provenance data contains the data recorded by a workflow engine during a workflow execution. It identifies what data is passed between services, which services are involved, and how results are eventually generated for particular sets of input values. Data associated with a particular service, recorded by the service itself or its provider, is also stored as provenance information. Such data may relate to the accuracy of results a service produces, the number of times a given service has been invoked, or the types of other services that have made use of it [2].

One of the unexplored applications of provenance is exploiting it for the purpose of learning. A large store of the previous executions of services and workflows, as well as their specifications, provide an appropriate data set for learning and knowledge discovery. The provenance data can be explored using data mining and pattern recognition methods to discover the patterns of interest in the data. The store is also a suitable source for learning probabilities. Therefore, probability learning methods can be used to produce the required parameters for the probabilistic decision making processes. As the provenance data is recorded at regular intervals, and consists of values and events that are changing with time, we believe time series mining methods [1] are a suitable choice for evaluating and describing the changes that occur in the data.

Applying learning and knowledge discovery methods to provenance data can provide rich and useful information on workflows and services. Therefore, the chal-

lenges with workflows and services are studied to discover the possibilities and benefits of providing solutions by using provenance data. Previously, large amount of research has been done to target workflow challenges such as composition, pattern discovery, service selection, and process refinement. Workflow composition and selection methods require a description of resources and Quality of Service (QoS) specifications as well as well-defined inputs and outputs. These descriptions are usually presented in the service ontologies provided in service registries. As the provenance store keeps the specification of services such as input or output or service description, it can be regarded as a large informational registry providing the chance of intelligent composition and service selection using previous experiences. Among the workflow issues and challenges, workflow analysis and evaluation, which mostly includes QoS assessment and trust measurements, is the least-attended problem. Provenance provides a suitable resource of information for performing analytical evaluation on data. Discovering workflow patterns has been previously studied using event logs, which provide a very small amount of data for learning the workflow models, while provenance provides a rich knowledge base for extracting hidden and unknown models [3].

The remaining sections of this book chapter are organized as follows: in Sections 2 and 3 the motivation and requirements as well as the multi-functional provenance architecture[1] is described. Section 4 outlines how workflow trust can be evaluated using the Hidden Markov Model, in Section 5 we discuss the procedure followed for assessing the HMM probabilities, and in Section 6 the implementation details of the model are provided. Section 7 presents a case study, as well as the stationary assumption of the model is investigated and some experiments are performed to compare the NSHMM trust evaluation results with HMM. In the final section the conclusion and future work is given.

2 Motivation and Requirements

A service-oriented architecture provides an environment in which services are shared among distributed systems. Potentially, thousands of services are available, which can be discovered or combined dynamically through appropriate mechanisms for the purpose of workflow selection, composition, or refinement. Thus, current major issues regarding workflow and services can be summarized to service composition and selection, workflow model extraction, refinement, and evaluation. In literature, these problems are targeted via semantic descriptions of services and event logs. In this section, we are going to discuss the knowledge requirements of each problem, and will argue how provenance data satisfies these requirements and provides a suitable platform for improving as well as optimizing the quality of the solutions to these problems. Workflow composition and selection methods require an expressive language that supports flexible descriptions of models and data

[1] these two chapters have been partly published in [2]

to facilitate reasoning and automatic discovery and composition. Therefore, they mostly exploit the semantic descriptions of services as well as their QoS specifications from service repositories or service providers to perform the composition or selection. In [7], the authors discuss the requirements for workflow composition. These requirements can be summarized as follows:

- Workflows must be described at different levels of abstraction that support varying degrees of reuse and adaptation. It is important to mention that this requirement is based on the fact that workflows can often be created by re-using existing workflows with minimal changes.
- Expressive descriptions of workflow components are needed to enable workflow systems to reason about how alternative components are related, the data requirements and products for each component, and any interacting constraints among them.
- Flexible workflow composition approaches are needed that accept partial workflow specifications from users and automatically transform them into executable workflows.

In order to satisfy these requirements, the authors consider three stages for the creation of the workflows, which include: defining workflow templates, creating workflow instances that are execution independent, and creating executable workflows. The three requirements mentioned can be satisfied through provenance data. In [5], the authors argue that a robust provenance trace provides multiple layered presentation of provenance. Thus, a layered architecture and engine for automatically generating and managing workflow provenance data is considered in provenance systems. As a result, provenance data can be used for interpreting the services and datasets of the workflows. Provenance creation is performed by following a layered approach that fulfills the requirements of the workflow composition process. The first layer of the architecture represents an abstract description of the workflow that consists of abstract activities with the relationships that exist among them. The second layer provides an instance of the abstract model by presenting bindings and instances of the activities. The third layer captures provenance of the execution of the workflow including specification of services and run-time parameters. The final level captures execution time specific parameters including information about internal state of the activities, machines used for running, status and execution time of the activities.

As the execution time specific parameters are also gathered in provenance stores, provenance data also includes the QoS specifications of services. Thus, service selection solutions can be applied to this data in order to automatically select appropriate services that provide some QoS requirements. Service providers may not be trustworthy enough to deliver the services based on the agreed-on QoS. On the other hand, the *validity period* of the agreement might have come to an end and no agreement updates might have been made afterwards. The ontological QoS specification of service providers are updated periodically while there might be many requests in each period. In case the QoS guarantees change during a period, the providers will not be able to satisfy the agreed-on thresholds. Or the service provider might not be able to provide the specifications at all. Using the history of previous executions,

the provided QoS overcomes the inconsistencies between the guaranteed and delivered QoS values of services to some extent by providing an estimate of the QoS parameters of the services with regard to time.

Most research on workflow systems focus on prediction, tracking and monitoring of workflows, and not on the evaluation of these processes. Few research efforts which studied the evaluation component of workflows, investigated a very narrow research problem aimed to improve the performance or fault tolerance of workflow systems [6]. As the provenance information maintains the records of previous execution details of workflows, it provides the facility to analyze, assess, and evaluate the behavior of a workflow as well as its performance. The performance of a workflow, its trustworthiness, improvements, and its future trend, etc. can be analyzed and evaluated through provenance data.

Workflow mining discusses techniques for acquiring a workflow model from a workflow log. Workflows can be investigated from many perspectives: functional, behavioral, informational, organizational and operational. In case of the behavioral perspective, which looks at control flow, workflow mining is done by following the order in which events for tasks are stored; for the informational perspective which looks for data flow, usually inputs/outputs are being used; in case of the organizational perspective, participants of tasks and their roles are being discovered. The workflow mining methods use the event-logs for discovering the patterns and mining the workflows, which keep track of a very small amount of information. The information provided in event logs is not enough for mining workflows with regard to all the mentioned workflow perspectives while much stronger reasoning and mining can be done over the data presented in workflow provenance.

To improve the efficiency of the composition and selection processes, previous executions of workflows and services can be used to augment these processes with more intelligence during the composition or selection. The feedback learned through previous runs secure the composition (or selection) from services that either do not have available resources, or do not satisfy the promised trust levels at a particular time. In case of the composition, the feedback of previous runs of the composed process will also be analyzed later to discover the possible deficiencies that might exist in the composed model.

As more provenance information is gathered, the extracted workflow process models are refined over time and the structure is geared to improve the efficiency with regard to changes in the data. These variations might include updates of the most frequently chosen paths, or assigning/changing the weights of the links in the model with regard to the rate of usage in time. These types of augmentations in the model also facilitate the process of refining or repairing a workflow model.

Since the provenance information of the same executions might provide the intermediate data generated by a process, the processes can be reduced by removing existing services, or replacing the parts, which cannot be executed with other parts, by looking for a more optimal path in the extracted workflow model with regards to the weights of the connections.

As mentioned earlier, the history of previous executions of workflows and services satisfies the requirements of addressing the discussed challenges. Apart from

the requirements, it was discussed that the provenance data augments the challenges with more efficiency, and reliability. Thus, there is a need for an architecture that facilitates addressing and solving all these aforementioned issues by exploiting the provenance data.

3 Architecture

In this section, the multi-functional architecture discussed earlier is presented along with its components. Figure 1 outlines the architecture. The structure is composed of 5 components that cooperate together along with the provenance store to provide different functionalities. The responsibilities of each component, the way components collaborate to provide the promised functionalities, and the approach taken to achieve the goals of the components are discussed.

Workflow Model Extraction and Discovery Component: This component is responsible for extracting the workflow pattern and associations that exist among the relevant workflows previously run and executed. Two workflows are considered relevant if they are in the same area of interest. The extraction component discovers the hidden connections that might exist among services and were not known beforehand. It generates a policy graph of the relevant services with edges representing the associations between them. The output is an optimal policy graph including all possible paths that could exist between the services of similar functionality. The extracted policy graph can be used later for the purpose of workflow construction and repair. The component is also able to receive a workflow pattern, and look for the same pattern sequence in the store to discover if there is any information regarding its previous executions in the provenance store.

Workflow and Service Evaluation Component: Evaluating workflows and services in terms of trust and quality is an important and less studied topic in the area of workflows. Workflows need to be assessed and analyzed to discover how trustworthy the composition of services are, therefore, in case the trust given by a workflow is not satisfactory, the workflow sequence can be repaired and improved. Another responsibility of this component is to identify the points in time at which a significant variation in trust occurs. This information can help us in identifying the parts of the workflow that are not providing the promised or required trust levels. Similar to workflows, the services are evaluated by this component. Large fluctuations of the QoS values of services are investigated to predict when in the future the service will not support the promised QoS requirements. Based on the previous executions, this component is also able to predict which services are going to be executed and in case the results of another instance of the same service are available, the process of workflow execution can be improved by exploiting those results. Apart from the trust assessment, the performance of the workflow is evaluated in terms of resource usage, and total time elapsed from the submission to completion.

Workflow Repair and Refinement Component: In case a workflow does not provide the required trust level, or it cannot be executed due to lack of available

services, the workflow needs to be repaired or refined. The repairment/refinement component takes advantage of the extracted policy graph of the workflow along with the assessment results of the evaluation component. The policy graph is traced to find a path that can replace the defective part of the workflow. The defective path is either inefficient due to lack of trust provision, or cannot be executed any longer because of unavailable services. In case a service is predicted to not provide the promised non-functional requirements, the service is replaced by another service or services to provide a similar functionality.

Workflow Composition and Generation Component: Composing a set of services using provenance data is a very useful exploitation of the provenance store. The stored specifications of services and their states provide the facility of composing the services automatically. On the other hand, having the previous history of executions, provides the data, which is essential for learning, therefore, the composition will be done in a more efficient way by exploiting the provenance data. This component receives the requirements and composes a workflow dynamically by taking advantage of the service specifications provided in the store. Previous execution of workflows enables the composition to be more robust as it exploits the evaluation results of services and workflows to generate a well-designed workflow process.

Workflow Service Selection Component: The problem of selecting a set of concrete services that provide the required QoS specifications for a complete abstract workflow is referred to as abstract workflow service selection problem. The provenance data can be exploited to speed up this task. In order to find the set of concrete services that match a single abstract service, service registries are looked at and matchmaking algorithms are applied to discover the matching services. The service discovery phase is much simpler if provenance data is used. Previous ex-

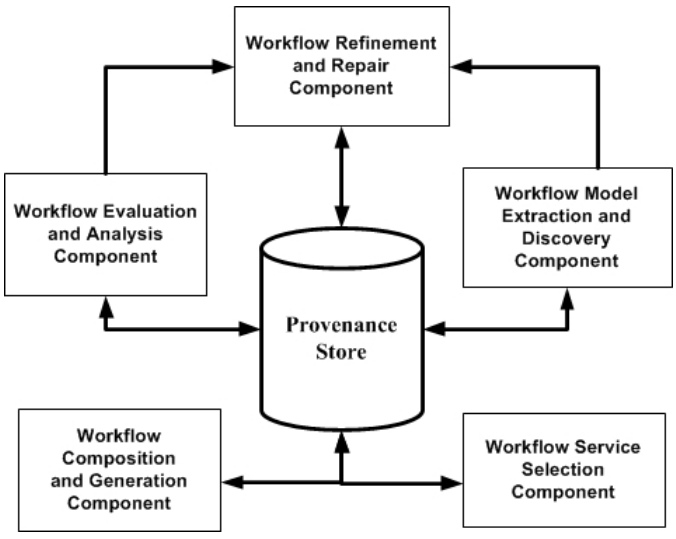

Fig. 1 Architecture

ecutions of workflows along with the workflow templates simplify the process of service discovery for a simple query. The set of suitable concrete services for the abstract workflow can then be selected more optimally by using the selection mechanisms along with the evaluations of previous executions.

4 Hidden Markov Modeling for the Evaluation of Workflow Trust

In the remainder of this book chapter we want to focus on the workflow evaluation component of the architecture.

Execution of a sequence of services requires much more resources and time in comparison to a single service. Thus, if a workflow is not very reliable, many resources and time will be wasted; since the results of the workflow can not be trusted. Therefore, it is important to be able to evaluate the trust of a workflow to find the degree of reliability of the workflow and its results. This also helps to decide whether the workflow needs some refinement and whether less trustful services should be exchanged with more trustful ones.

Having the trust value of each service, allows to evaluate the overall trust value of a sequence of services, i.e. a workflow. Therefore, we can determine the amount of trust that can be placed on the overall workflow as well as the results and datasets generated during the workflow execution. There are very few approaches addressing the subject of workflow trust evaluation. One approach uses a decision tree model, which is presented in [19]. In this paper, a decision tree is built out of a question sequence that will help in assessing the trust that can be associated with the data produced from a process. The root node asks about the trust of the workflow and has three child nodes, evaluating the trustfulness of services, data and the workflow process. Each child node has a sub-tree representing a set of yes/no questions. The decision making process starts with one child node, traverses its sub-trees and continues to the next child node. This procedure is followed continuously until all the sub-trees are investigated. The result of the investigation is either a yes or no, determining whether the workflow can be trusted or not. This work has been extended and an important shortcoming of it, the crisp result, has been addressed in [20]. Therefore, the outcome of each analysis node of the trust decision tree is mapped to a fuzzy membership function. Later, these values are combined together using fuzzy inference rules.

However, all the current solutions lack accuracy, automation, and reliability. They are based on a decision tree model with categorical nodes that have been designed by the developers. The decision nodes of the tree are simple sets of questions regarding the user's views or behaviors toward service, data or process trust. Besides, the trust value of each service or data is not considered separately, but instead the overall trust level of services is involved in the decision making process.

We propose a new approach for the evaluation of trust of workflows, which is based on a statistical model named Hidden Markov Model (HMM). Rather than

traversing a set of question nodes, in our model, the trust will be assessed by solving a set of mathematical equations that describe the behavior of the workflow trust in terms of random variables and their probability distributions. Thus, our method is more accurate in comparison to the previous approaches and will support automation.

A HMM is a probabilistic process over a finite set of states, where each state generates an observation. Given a HMM, and a sequence of observations, the probability of the observation sequence given the model can be evaluated. It is also possible to discover the hidden state sequence that was most likely to have produced the observation sequence. Another type of inference on HMMs can estimate the HMM model through training examples and learning methods.

HMM has become the method of choice for modeling stochastic processes and sequences in applications such as speech and handwriting recognition [8], computational molecular biology [9], natural language modeling [10], etc. In this work, HMM is used for the purpose of workflow trust evaluation.

In order to be able to assess the proposed HMM model, probability learning algorithms like Maximum Likelihood (ML) or Expectation Maximization (EM) learning techniques are used along with provenance data. Provenance is one of the growing demands in distributed service oriented environments, which supports the systems with documentation of the origin and the processing steps of data that is part of a workflow execution process. It also provides explanations about which, how and what resources and services were used to produce that data, and is referred to as provenance data that is captured and stored in provenance stores for the purposes of reasoning, validation and re-execution. A provenance store provides the necessary information that is exploited for the purpose of estimating HMM probabilities.

Many approaches have been proposed to improve the predictive power of HMM in practice. For example, factorial HMM [12] is proposed to decompose the hidden state representation into multiple independent Markov chains. In speech recognition, factorial HMM can help in representing the combination of multiple signals. Hierarchical HMM [13] is another method that facilitates the inference of correlated observations over long periods in the observation sequence via higher level hierarchy. However, from the essential definition of HMM, there are other ways to improve the predictive power of HMMs. One approach is to relax the stationary hypothesis of HMMs and make use of time information. To investigate this further and observe the behavior of our model with regard to the non-stationary assumption, the workflow trust has also been evaluated using the Non-Stationary HMMs (NSHMM).

5 Methodology

The notion of trust of an enacted workflow is an important issue in distributed service oriented environments. Trust evaluation aims at contributing in the discovery of how trustful the results of a workflow are. It also helps the optimization of composite

service executions. In this section, we are going to first present how the workflow trust can be evaluated using hidden Markov modeling. Later, we explain how the model can be assessed by taking advantage of the previous history of the execution of workflows.

A HMM is a statistical model that can be considered as the simplest kind of Dynamic Bayesian Networks (DBNs). The system that is being modeled according to HMMs is assumed to be a Markov process with unknown parameters. Markov processes are an important class of stochastic processes that are governed by the Markov property. The Markov property states that the future behavior of a process given its path only depends on its present state. The HMM model basically consists of two sets of variables: state variables and evidence variables, which are also called the observations. The state variables are the hidden variables that change over time; while the evidence variables are the observable variables that are known in advance at each time step. The challenge is to determine the hidden parameters from the observed ones.

Figure 2 shows a simple first order HMM. The state variable x_t is a hidden variable at time t and can have a value from x_t the domain of x. The random variable y_t denotes the observable parameter at time t. From the figure, it can be seen that the value of the hidden variable at time t, i.e. x_t, depends only on the value of the hidden variable x_{t-1}, and other previous parameters have no influence on it. This property is referred to as the first order Markov property.

In order to model the workflow trust evaluation as a HMM, the state and observable variables are mapped as follows:

- Tr_t: the trust state variable, represents the state of the trust of the workflow at time t.
- S_t: the evidence variable represents the service that is being executed at time t.

Figure 3 depicts a simple linear workflow and the correspondent HMM, modeled to evaluate the trust level of the workflow. As it can be observed form the figure, the state of the trust of the workflow at the beginning (Tr_0) is only determined by the evidence variable observed at that time (x_t). For the following time steps, the state of the trust of the workflow can be determined by investigating the state of the workflow at the previous time step, and observing the service that was executed at that time.

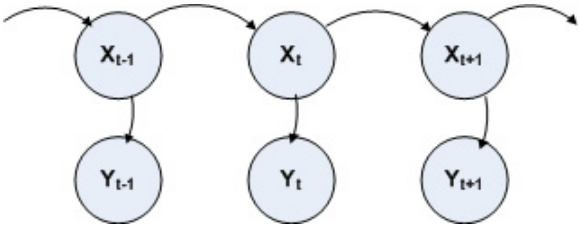

Fig. 2 Basic HMM.

In theory of HMMs, some assumptions are made for the sake of mathematical and computational tractability. Here we present how these assumptions can be applied to our model:

1. The Markov assumption: It is assumed that the next state is dependent only upon the current state. This is true in case of our model, as the state of the trust of the workflow at each time only depends on the state of the trust at the previous time and not the other prior states.
2. The output independence assumption: This is the assumption that the current observation is statistically independent of the previous observations. In case of our model, the service at time t is independent of the previous services.
3. The stationary assumption: This assumption is based on the fact that the transition probabilities between the states are independent of the actual time at which the transitions take place. In case of the workflow trust problem, we can not say that transition probabilities are completely independent of time. We suppose that this assumption will be true for our model since we can take the average of the state transitions of all times and have one set of state transition probabilities for the overall time period. In order to investigate this further, later in the paper, we will observe the behavior of the model by relaxing this assumption and having a non-stationary HMM.

Having defined the HMM and described how the HMM parameters and assumptions can be mapped to the workflow trust evaluation parameters, we will now clarify how this model can be exploited for the purpose of trust evaluation.

As mentioned earlier, different kinds of inference can be done on HMM structures. These include methods for computing the posterior distribution over the current, future, or a past state, or finding the sequence of states that is most likely to have generated those observations. Filtering or monitoring is the task of computing the posterior distribution over the current state, given all evidences and observations to date. The following probability expresses filtering inference:

$$P(X_t \mid y_1, y_2, ..., y_t) \tag{1}$$

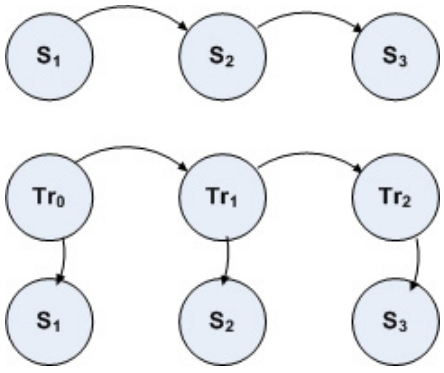

Fig. 3 A sample workflow and the HMM for workflow trust evaluation.

Using the filtering model, the probability of the state of the trust at the final state of the workflow can be roughly estimated given all the observations, which are the services seen so far. Therefore, for the case of the trust evaluation, the following probability should be assessed:

$$P(Tr_2 \mid s_1, s_2, s_3) \tag{2}$$

for different possible trust state levels. Evaluation of the above probability provides us with estimations of probabilities for different trust levels at time t_2. In this work, the state of the trust will be evaluated at three different levels of *High*, *Medium* and *Low*. The work can later be extended to support further trust levels.

5.1 Trust Model Assessment

In order to be able to compute the filtering inference, two other probabilities should be assessed beforehand. These probabilities are referred to as state transition probability and sensor probability. The state transition probability is defined as the probability of being in the next state given the current state, i.e. $P(x_t \mid x_{t-1})$, which in our case is the probability of being at a trust level at time t given the level at the previous time, i.e. $t-1$. The sensor probability is defined as the probability of the observation at time t, which is the service that was executed at time t, given the different level of trustworthiness of the workflow at that time. To assess the state transition or sensor probabilities, the ML or EM learning algorithms are utilized along with the provenance data.

In service-oriented environments, great numbers of workflows are executed to perform computational and business experiments. The workflow activities are run repeatedly by one or more users and large numbers of result data sets in the form of data files and data parameters are produced. As the number of such datasets increases, it becomes difficult to identify and keep track of them. Besides, in these large scale scientific computations how a result dataset is derived is of great importance as it can specify the amount of reliability that can be placed on the results. Thus, information on data collection, data usage and computational outcome of these workflows provide a rich source of information. Capturing this information, which is regarded as provenance information, is a significant advantage of using workflows. Provenance information facilitates data dependency determination, workflow result validation, efficient workflow re-executions, error recovery, etc. [14]. Provenance also enables users to trace how a particular result has been arrived at by identifying the aggregation of services that produces such a particular output. This data can provide us with the history of previous execution details of workflows. In this work, we are exploiting the provenance data to learn the HMM probabilities.

5.1.1 Assessment of Transition Probabilities

In order to assess the transition probabilities, the trust state transitions, i.e. $P(Tr_t \mid Tr_{t-1})$, should be computed for all pairs of workflow services that are being executed in sequence. Having a large provenance record of the previous executions of workflows, we will be able to learn the transition probabilities by applying the ML method on the provenance data.

ML learning is a data analysis approach for determining the parameters that maximize the probability (likelihood) of the sample data, which is trust state transitions in this case. From a statistical point of view, the method of ML is considered to be robust and yields probabilities with good statistical properties [15].

To assess this probability using the ML method, we determine the number of each trust state transition with regard to the total number of transitions of that state. The transition probability estimation for our model is computed based on Equation 3:

$$P(Tr_t = j \mid Tr_{t-1} = i) = \frac{n_{ij}}{n_i} \tag{3}$$

where n_{ij} denotes the number of transitions from trust level i to trust level j, and n_i denotes the number of transitions from trust level i. For example, for the sample workflow in Figure 3, which was composed of three services, the trust state transition from high to low will be computed by first determining the number of high to low transitions for the service pairs (s_1, s_2) and dividing it by the number of times the service s_2 had low trust level. The same will be done for the pair (s_2, s_3). The average of these values represents the transition probability from high to low.

It is important to mention that the same pair of sequential services might be repeated in several workflows, and the transition probabilities for these services will be learnt without considering specific workflows. The average of all these probabilities will denote the final transition probability for these pairs of services.

Assessment of Sensor Probabilities

To assess the sensor probabilities for each time instance t, the probability of observing an evidence variable given the state at that time should be computed. Therefore, we should compute $P(S_t \mid Tr_t)$, which again will be learnt by utilizing the ML method and the provenance data.

For this purpose, the number of times the trust state of service instance S_t was at each trust level is estimated. This value is divided by the total number of times any service was at that trust state. As before, the provenance history of the workflow will be used. Equation 4 represents the assessment of the sensor probabilities for our model:

$$P(S_t = s_t \mid Tr_t = j) = \frac{n_{st\,j}}{n_j} \tag{4}$$

where $n_{st\,j}$ denotes the number of times being in state j and observing service s_t, and n_j denotes the number of times being in state j.

Assessing the Trust Level

Having assessed the sensor and transition probabilities, we will be able to assess the filtering model of HMM and therefore evaluate the workflow trust using Equation 5:

$$P(Tr_t \mid S_1 = s_1, S_2 = s_2, ..., S_t = s_t) = \propto P(S_t = s_t \mid Tr_t) \sum_t (P_{Tr} \mid P(Tr_{t-1}) \\ P(Tr_{t-1} \mid S_1 = s_1, S_2 = s_2, ..., S_{t-1} = s_{t-1}) \quad (5)$$

The probability of $P(Tr_{t-1} \mid S_1 = s_1, S_2 = s_2, ..., S_{t-1} = s_{t-1})$ is computed recursively. Equation 6 evaluates the probability of different trust levels at time t having observed the services the workflow is composed of until that time.

As discussed, for the purpose of assessing the probabilities, the ML learning algorithm is utilized in this work. This is based on the assumption that the provenance data does not include a large amount of missing data. To be able to find the probabilities in case of missing data, the EM learning algorithm can be used. The EM algorithm is an efficient iterative procedure to compute the ML estimate in the presence of missing or hidden data. Using this algorithm, we first predict the missing values based on assumed values for the parameters. Later, these predictions are used to update the parameter estimates. The sequence of parameters converges to ML estimates, and EM implicitly averages over the distribution of the missing values.

5.2 Cases with Dynamic or Parallel Sections

The presented trust model is compatible for workflows which contain not only sequential but also parallel sections in the workflow. In case of non-sequential workflows, a sequential workflow is extracted from them by selecting one of the subsections of each parallel section according to a policy, and replacing that parallel subsection with the selected subsection. Starting with the deepest parallel subsections, a subsection is chosen for each section by first applying the HMM model to all the parallel sub-sections of that section, and then the trust level probabilities of the subsections are compared with each other. For each section, the subsection that has the lowest trust level is selected and the parallel section is replaced by that subsection. By following this policy for all the parallel sections, the workflow is transformed to a sequential workflow, and finally the HMM model is applied to assess the trust level.

It is important to mention that as the proposed approach exploits provenance information to get an assessment of the QoS values, it works for the static scenarios. In case of workflows with services for which there is no history in the provenance store, the online QoS values presented by the service provider are used for assessment.

6 Implementation

As mentioned earlier in this work, the trust of each service instance is categorized into three levels of *High*, *Medium*, and *Low* and can be evaluated by aggregating the QoS parameters of the service. These QoS parameters can include status, availability, reliability, execution time, reputation, etc. The trust value is usually determined by assigning a weight to each parameter and the summation of the multiplication of the parameters by their weights results in the final trust value. As in our current model we are concerned with trust levels rather than trust values, we determine the level of the trust with regard to the level of the QoS parameters.

In our implementation, we have considered the QoS parameters of status, reliability and availability. The QoS parameter *status* is a binary value that represents the status of the execution of the service. A value of 1 describes that the service was executed successfully and a value of 0 reports unsuccessful execution. The QoS parameter availability presents how available a certain service and its data are, while reliability denotes the degree we can rely on the processing and the response time of the service. Both parameters have a value in the range of [0,1].

In order to decide about the trust level of each service using these parameters, we followed a table model, Table 1, in which the level of all QoS parameters of availability and reliability in conjunction with the status of the execution determines the level of the trust. The table is referred to as the trust level decision table throughout this book chapter. A sample row in this table represents the associated trust level in combination with the discussed QoS parameters. For example, LL1 denotes that the level of the reliability and availability of a service is *Low*, and the status is 1. According to the table, the trust level of the service is assessed as *Low*.

The levels of *reliability* and *availability* of the services are determined according to a set of pre-determined range levels. For the examples and experiments provided in this book chapter, the following range table (Table 2) was used.

As was discussed earlier, the probabilities are assessed by applying learning methods over the provenance data. For the purpose of learning, we implemented a provenance store in MySQL [18] including tables for storing the information of workflows, services, workflow instances, and workflow sequences. The provenance data is then generated by a random workflow generator implemented to produce instances of a workflow. The generator asks for the following parameters as input:

- N_s: the number of services the workflow should be composed of.
- N_w: the number of previously executed instances of the workflow.

In order to assess the HMM, we followed the matrix algorithm which describes the sensor and transition models in form of matrices. The transition matrix denoted by T is a $m \times m$ (in our case 3×3) matrix where m is the number of possible states. The probability of a transition from state i to state j is denoted by the entry T_{ij}:

$$T_{ij} = P(Tr_t = j \mid Tr_{t-1} = i) \tag{6}$$

Table 1 Trust level decision table, L, M, and H denote Low, Medium, and High.

Trust	Reliability, Availability, Status
L	LL0
L	LL1
L	ML0
M	ML1
L	HL0
M	HL1
L	LM0
M	LM1
L	MM0
M	MM1
L	HM0
H	HM1
L	LH0
M	LH1
M	MH0
H	MH1
M	HH0
H	HH1

Table 2 Range Level of the QoS parameters Availability, and Reliability.

Trust Level	Low	Medium	High
Availability	[0,0.3]	(0.3,0.7)	[0.7,1]
Reliability	[0,0.3]	(0.3,0.7)	[0.7,1]

which, as discussed, will be evaluated using the generated provenance data along with the trust level decision table (Table 1), QoS parameters range level (Table 2) and the ML algorithm.

The sensor model is also put into matrix form. For each time step t, a diagonal matrix, O_t, is constructed whose diagonal entries are given by the values $P(S_t \mid Tr_t = i)$, with the other entities set to 0.

Now, to accomplish the filtering inference and represent the forward messaging in HMMs using the matrix model, Equation 7 is applied recursively:

$$f_{1:t+1} = \alpha O_{t+1} T^T f_{1:t} \tag{7}$$

where α is the normalization factor. The result is a one column matrix denoting the probability of the trust level of the workflow for all the different possible levels.

6.1 Verification of the Model

Our approach is verified by a comparison done with the Viterbi algorithm [17], which finds the most likely sequence of hidden states that result in a sequence of observed events. For the verification, different observation sequences of different sizes were generated and the most likely sequence of underlying hidden states that might have generated those observation sequences was produced by applying the Viterbi algorithm. Having compared the resulting hidden states of the algorithm with the real hidden states, we received identical results. Therefore, this verifies that the HMM modeled for the purpose of workflow trust evaluation and the way the probabilities were assessed is valid.

7 Case Study

In this section, we present a workflow scenario and describe how its trust can be evaluated using the presented model. The sample workflow is the process of knowledge discovery in databases which is referred to as KDD process [16]. The KDD process is composed of four services for data selection and cleaning, data transformation, data mining, and data interpretation. Figure 4 shows the process.

The following assumption is made. A distributed service-oriented environment is sharing services for the purpose of knowledge discovery, and that a workflow is executed using four different services shared by service providers in the environment each having different QoS values, and therefore, different trust estimations. Using the workflow generator, the above workflow was defined and 50 execution instances were generated, representing the provenance data. Table 3 shows the average of the QoS parameters of those instances.

Table 3 The average of the values of the QoS parameters generated for the scenario.

QoS Parameter	Reliability	Availability	Status
Data Selection	0.58	0.59	0.8
Data Transformation	0.7	0.7	0.88
Data mining	0.34	0.34	0.82
Interpretation	0.84	0.84	0.82

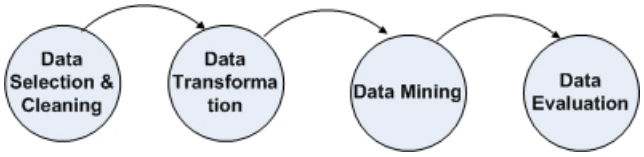

Fig. 4 A sample workflow scenario - KDD Process.

The QoS parameters *availability* and *reliability* were generated in the range of 0.3 to 0.9, which mostly covers the medium and high trust levels. The status of the execution was set to zero in less than 20% of the cases. It is important to emphasize that according to the trust level decision table (Table 1) the state of the trust of a service instance is evaluated as *Low* if its status is zero. The reason for this decision is that if a service does not complete its execution successfully, that service instance should not be trusted at all. Therefore, we evaluate the trust as low regardless of the instance's level of *reliability* and *availability*.

In the next step, the transition matrix is built by learning the probabilities from the generated provenance data. Given the data, the transition matrix T, of the above example was estimated as given in Figure 5.

L, M, and H represent the trust levels *Low*, *Medium* and *High*. An entry T_{ij} denotes the transition probability of being transferred from trust level i to j. For a better understanding, the state transition diagram is also provided in Figure 6, which is the same as the transition matrix but presents it in a graphical view which is easier to follow.

Having learnt the transition matrix, the forward algorithm starts with assessing the sensor probability at the first time step and forwards this message along with the transition messages to the next time step. This process of forwarding messages continues until the last service is observed, and therefore the overall trust of the

	L	M	H
L	0.21	0.22	0.57
M	0.184	0.316	0.5
H	0.17	0.83	0

Fig. 5 Transition matrix of the example

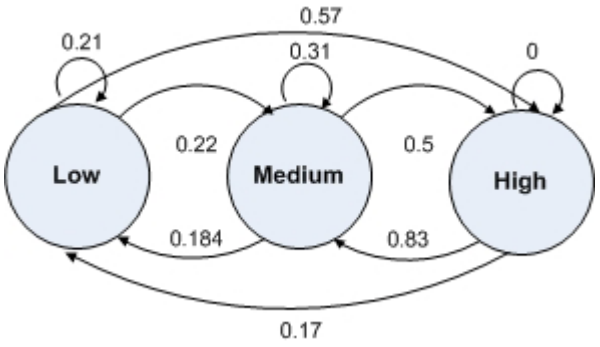

Fig. 6 The state transition diagram showing the transition probabilities for the above example learnt through ML method.

workflow is evaluated. It is important to mention that the prior belief about the trust state probabilities, i.e. the initial state probabilities, is considered equal for all the three possible states and was set to 0.33 for all the trust levels.

To investigate the behavior of the filtering method and observe the trust level probabilities estimated at each time step is provided in Figure 6. The figure shows how the trust state probabilities change over time during the HMM assessment for the discussed example.

It can be observed that the trust state is evaluated as *Medium* after observing the first service, it then heads toward *High*, then again *Medium* and finally the trust level is evaluated as *High*.

Taking a look at the average values of the QoS parameters of each service explains the behavior of the model. According to the QoS range evaluation table (Table 3), the trust level of the first service, which is the data selection service, can be evaluated as *Medium*. The trust level of the third service is also evaluated as *Medium*, and the trust level of the second and the fourth service is estimated as *High*.

The explanation above and the transition matrix shown in Figure 5 describe the reason behind the path taken in Figure 7. The path shows the route between the trust levels with the highest probabilities at each time step. The transition probabilities with large probability values include transitions from *High* to *Medium*, *Low* to *High*, and *Medium* to *High*. The evaluation process starts with the first service which has an average of *Medium* trust level. As the transition probability of *Medium* to *High* is the largest, this leads the state of the trust toward *High*. Being in state *High* and having observed a service with *High* trust level leads the trust level toward *Medium* as the largest transition probability from *High* is the one toward *Medium*. The rest of the transitions can be explained in the same way.

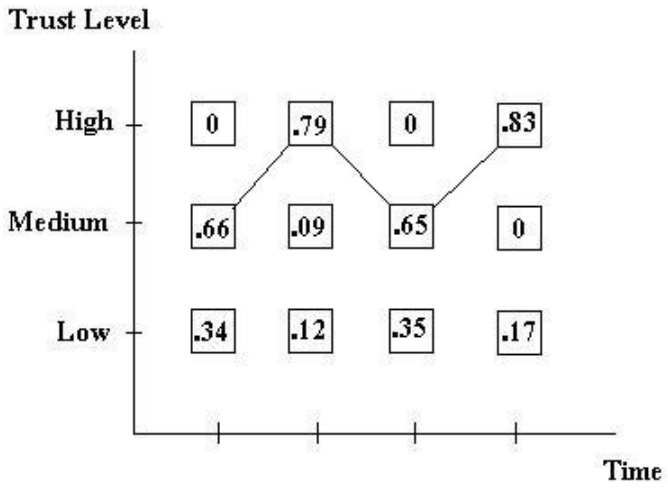

Fig. 7 The change of the trust state probabilities over time using a HMM.

It should be considered that there is always a less than 20% probability for a low trust state to be chosen for all the services. Because as discussed earlier, the status of the executions of services were randomly set to zero in almost 10 to 20 percent of the cases. Therefore, the final trust level probabilities will have a 10% low level probability on average.

7.1 Investigation of the Stationary Assumption

It was mentioned earlier that one of the assumptions of HMM is the stationary assumption. In order to follow this assumption, the transition probabilities were assessed by taking the average of the transitions between each pair of services to have the same state transition matrix at all times. As this assumption can not be verified completely in case of the workflow trust problem, this section investigates how the model will behave if we relax this assumption and transition probabilities are considered time-dependent. To achieve this goal, the transition probabilities are computed separately for each time step.

In the theory of HMMs, it is assumed that state transition probabilities are independent of the actual time at which the transitions take place. This assumption can be mathematically presented as:

$$P(x_{t1+1} = j \mid x_{t1} = i) = P(x_{t2+1} = j \mid x_{t2} = i) \tag{8}$$

for any t_1 and t_2. Equation 8 states that the transition probabilities are constant over time which means that the probability of transition between different trust levels is the same for all times. Therefore, the Markov chain is described as *stationary* in the strictest sense. In general, it is possible to lift the constancy constraint and define the transition probabilities as a function of time. This model is referred to as the Non-Stationary Markov Model (NSMM) [11] and has a set of transition probability distributions that vary over time. This means that, given a state i, the probability of moving to another state j is a function of time. The time can be either absolute or relative. Equation 9 shows how the state transition function can be estimated:

$$P_{ijt} = \frac{C(i,j,t)}{C(i,t)} \tag{9}$$

where $C(i,j,t)$ is the co-occurrence frequency of state i and state j at time t and it can be estimated by counting the co-occurrence times of state i and state j at the t^{th} time. $C(i,t)$ is the frequency of state i at time t and can be estimated by counting the occurrence times of state i in the t^{th} time. And P_{ijt} is the transition probability between state i and j at time t.

In case of the workflow trust evaluation, the transition probabilities can be considered as a function of time since the probability of transition from one trust level to the other at time t depends on the services that are being executed at that time instance. Therefore, it is important to investigate the behavior of the model this time

using the NSHMM in order to observe the effect of the stationary assumption on the trust evaluation results.

In case of relaxing the stationary assumption for the workflow trust evaluation, the state transition probabilities were assessed separately at each time step and a transition matrix was built using the ML method along with the provenance data representing the history of the observations seen previously at those time steps.

Following the ML estimation method, the transition probability from state i to state j at time t will be assessed as follows:

$$Pt(Tr_t = j \mid Tr_{t-1} = i) = \frac{n_{ijt}}{n_{it}} \tag{10}$$

where n_{ijt} denotes the number of transitions from trust level i to trust level j at time t, and n_{it} denotes the number of transitions from trust level i at time t.

The non-stationary model was further implemented and the result of the same scenario studied in the previous section was investigated using the new model. It is observed that the trust state probabilities have not changed much as time elapses. The maximum trust level path follows the same routine with very little changes in the state probabilities at each time. The evaluation result of the NSHMM shows that the workflow can be trusted with a probability of 93%, while using the HMM this probability was 83%.

To investigate this further, we ran experiments using both models and compared their results. The experiments were done by creating workflows with 5 to 25 services in increments of 5. A previous execution history of 50 instances was randomly generated for each workflow in order to learn the sensor and transition probabilities. The average of the trust level probabilities was then computed. It was observed from the experiment results that for both models the distance between the same trust levels was equal in 96% of the cases.

Figure 8 represents the average trust level probabilities of the HMM compared to NSHMM. It can be observed that the differences are very small. In all the experiments, the level of the trust was estimated to be the same.

In order to determine whether the results of the two models are the same, we ran the paired T-test on the datasets of the two models. The T-test is a statistical test that assesses whether the means of two groups of data are statistically different from each other. The result was a p-value of 0.78, which represents that the datasets are not significantly different from each other. The chart in Figure 8 and the T-test results both verify that the stationary assumption does not have a significant effect on the results of the trust level assessment, as both models provide estimations for the same trust levels with very little difference.

Experiments were done to compare both models in terms of the execution time and it was observed that while there is not large differences between the execution times, the execution time of the non-stationary model is larger. The reason for this observation goes back to the transition matrices that should be computed for each time instance separately while for the HMM with stationary assumption, the transition matrix is built once at the beginning by computing the average of all values.

8 Conclusion and Future Work

In this book chapter, a multi-functional architecture was described that addresses the current research issues of workflows and services using provenance data. The components of the architecture were described consisting of model extraction and discovery, workflow evaluation, workflow repair and refinement, workflow composition, and workflow service selection.

In addition, we focused on one component of the multi-functional architecture and put forward an approach for evaluating workflow trust level using hidden Markov models and provenance data. We discussed how the HMM assumptions can be applied to this problem, and we provided details on how the model can be assessed using the provenance data and maximum likelihood method.

In order to investigate the behavior of the model, we provided a workflow scenario and expressed how its trust level is evaluated using the proposed model. Furthermore, we presented how the Viterbi algorithm was used to verify the HMM. In order to verify the effect of the stationary assumption of HMMs for the trust evaluation problem, we investigated the results of applying the non-stationary hidden Markov model to our problem.

The two models were then compared with each other. It was observed that the same trust level was estimated by both models with a small difference in their probability values. Therefore, the stationary assumption does not have a significant impact on the trust evaluation results. The non-stationary assumption of transition probabilities seems to be more accurate in case of our model since the probability of moving from one state to the other at a time instance depends on the state of the two services

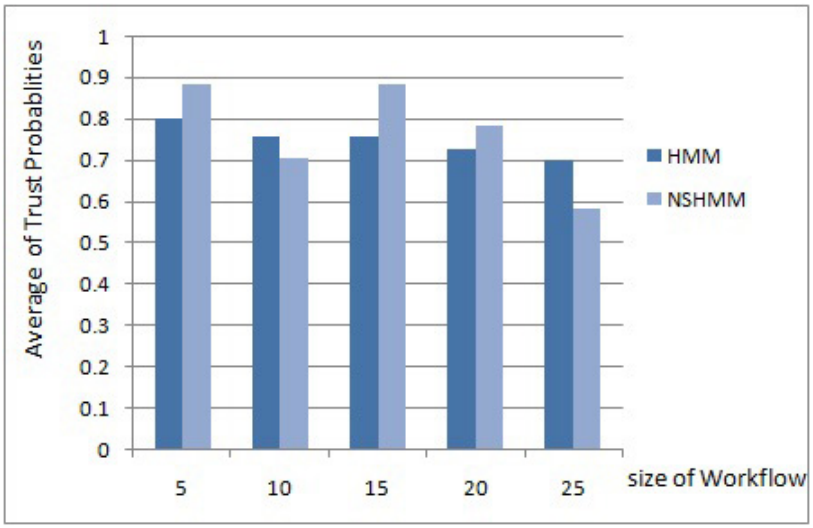

Fig. 8 Comparing the average trust level of HMM vs. NSHMM for 5 to 25 numbers of services with increments of 5.

that are being executed at those times. Thus, for this problem, it is better to consider the transition probabilities as time-dependent probabilities for more accurate results.

Future work involves performing a large number of experiments to evaluate the scalability and accuracy of the system, preferably with real data. Various experiments will be done for different workflow sizes, and the behavior of the system will be observed in response to larger workflows.

As the amount of provenance data affects the accuracy of the learnt probabilities, the reliability of the system will be evaluated considering different learning data. We will also consider incomplete data and experiments will be performed with EM learning to estimate the results in case of missing data.

The main concern of the current implementation was randomly generating a large amount of valid provenance data for many workflows, each having some common pattern with others. The future workflows ought to be realistic and consist of common services and patterns with reasonable provenance values and data from a number of executions. The model will be improved to also consider trust values of the workflow process and input data for the evaluations.

Furthermore, the fluctuation of trust with the Markov process needs to be investigated in order to discover the points at which the workflow lacks trustworthiness and should be refined. It is desired to automatically detect and replace less trustworthy services with trustworthy ones. This part of the work will be extended by learning the workflow patterns from the provenance data and substituting less trustful services or sections of the workflow with more trustworthy ones.

References

1. Brown B, Aaron M (2001) The politics of nature. In: Smith J (ed) The rise of modern genomics, 3rd edn. Wiley, New York
2. Naseri M., Ludwig S.A. (2010) A Multi-Functional Architecture Addressing Workflow and Service Challenges Using Provenance Data. Proceedings of Workshop for Ph.D. Students in Information and Knowledge Management (PIKM) in conjunction with the 19th ACM Conference on Information and Knowledge Management (CIKM), Toronto, Canada.
3. Gaaloul W., Baina B., Godart C. (2005) Towards Mining Structural Workflow Patterns. DEXA 2005, LNCS 3588, pp. 2433.
4. Altintas A., Lifecycle of Scientific Workflows and Their Provenance: A Usage Perspective (2008) In Proceeding of 2008 IEEE Congress on Services.
5. Kim J., et al., Provenance trails in the Wings-Pegasus system (2007) Concurrency and Computation: Practice and Experience, vol. 20.
6. Aiello R., Workflow Performance Evaluation (2004) PhD Thesis, University of Salerno, Italy.
7. Gil Y. Workflow Composition: Semantic Representations for Flexible Automation (2007) In: Workflows for e-Science, pp. 244-257.
8. Rabiner L. R. (1989) A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, pp. 257-286.
9. Krogh A., Mian S.I., Haussler, D. (1994) A Hidden Markov Model that finds genes in E.coli DNA, In: Nucleic Acids Research, vol. 22, pp. 47684778.
10. Jelinek F. (1985) Self-organized Language Modeling for Speech Recognition, IBM T.J. Watson Research Center Technical Report.

11. Bongkee S., Jin H.K. (1995) Nonstationary Hidden Markov Model, Signal Processing, vol. 46, pp. 31-46.
12. JingHui X., BingQuan L., XiaLong W. (2005) Principles of Non-stationary Hidden Markov Model and its Applications to Sequence Labeling Task, In Proceedings of the Second International Joint Conference on Natural Language Processing.
13. Fine S., Singer Y., Tishby N. (1998) The Hierarchical Hidden Markov Model: Analysis and Applications, Machine Learning, 32:41-62.
14. Altintas I., (2008) Lifecycle of Scientific Workflows and their Provenance: A Usage Perspective, IEEE Congress on Services 2008- Part I.
15. Verdonck F., Jaworska J., Thas O., Vanrolleghem P. (2001) Determining Environmental Standards using Bootstrapping, Bayesian and Maximum Likelihood Techniques: A Comparative Study. Analytica Chimica Acta 446, 429-438.
16. Fayyad M., Piatetsky-Shapiro G., Smyth P. (1996) From Data Mining to Knowledge Discovery: An Overview, in Advances in Knowledge Discovery and Data Mining, AAAI Press / The MIT Press, Menlo Park, CA, pp.1-34.
17. Forney G. D., (1973) The Viterbi Algorithm, Proceedings of the IEEE, Volume 61, Issue 3.
18. MySQL DataBase Software, www.mysql.com
19. Rajbhandari S., Wootten I., Shaikh Ali A., Rana, O.F. (2006) Evaluating Provenance-based Trust for Scientific Workflows. In Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid.
20. Rajbhandari S., Rana O.F., Wootten I. (2008) A Fuzzy Model for Calculating Workflow Trust using Provenance Data. In: Proceedings of the 15th ACM Mardi Gras conference.