

# Multi-level Monitoring and Analysis of Web-Scale Service Based Applications

Adrian Mos<sup>1</sup>, Carlos Pedrinaci<sup>2</sup>, Guillermo Alvaro Rey<sup>3</sup>, Jose Manuel Gomez<sup>3</sup>,  
Dong Liu<sup>2</sup>, Guillaume Vaudaux-Ruth<sup>1</sup>, and Samuel Quaireau<sup>1</sup>

<sup>1</sup> INRIA, 655 avenue de l'Europe, 38 334 Saint Ismier Cedex, France

<sup>2</sup> Knowledge Media Institute - The Open University Walton Hall, Milton Keynes, MK7 6AA,  
United Kingdom

<sup>3</sup> iSOCO, Pedro de Valdivia, 10, 28006 - Madrid

{Adrian.Mos, Guillaume.Vaudaux-Ruth, Samuel.Quaireau}@inria.fr,  
{C.Pedrinaci, D.Liu}@open.ac.uk,  
{Galvaro, Jmgomez}@isoco.com

**Abstract.** This paper presents a platform that aims at monitoring and analyzing large service-oriented applications executing on a very large scale. This is part of a vision of web-scale service utilization and management that is proposed by the SOA4All EU project. The paper shows how the platform obtains data from distributed runtimes and how it presents monitoring information at different levels of abstraction. They range from low-level infrastructure-related event details to high-level service and process analysis. Each level uses appropriate visualization techniques and widgets in order to convey the relevant information to the users in an efficient manner. The platform is under development and an advanced prototype is already available and described in the paper.

**Keywords:** monitoring, service-based applications, business processes, knowledge extraction, user interfaces for monitoring.

## 1 Introduction

SOA is largely still an enterprise specific solution exploited by and located within large corporations used mainly for integration. For instance, the current web only exposes around 28,000 traditional WS-based web services<sup>1</sup>. Nevertheless, as mobile devices and more efficient wireless communications facilitate ubiquitous computing, and as optical and broadband communication infrastructures expand, we expect the number of Web services to grow exponentially in the next few years.

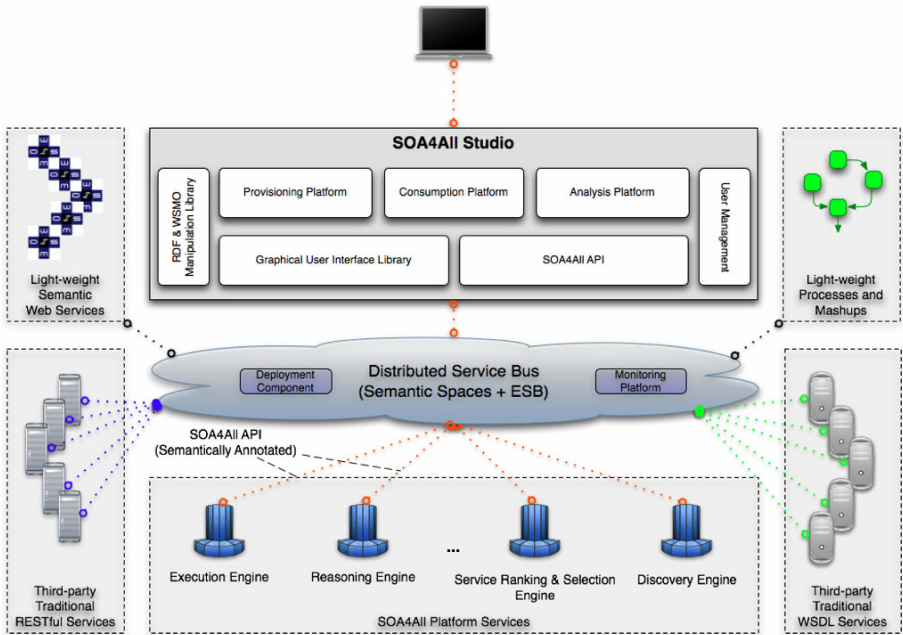
The main objective of the EU project SOA4All is to provide a framework and an infrastructure that help to realize a world where billions of parties are exposing and consuming services via advanced Web technology. An important part in realizing this vision is the capability to provide users with a good understanding of how services and processes perform at functional and non-functional levels. This is achieved

---

<sup>1</sup> According to seekda.com the number of WSDL services available online on March 04, 2009 was 27.813.

through the SOA4All Analysis Platform that aims to provide the SOA4All users with information that would help them understand the performance characteristics and usage patterns of the services and processes they share. This platform is part of the SOA4All Studio that groups all the user-facing functionality of SOA4All.

There are important scalability requirements on the overall SOA4All infrastructure and in particular on the Analysis Platform. For example, the monitoring and provenance tools should cope with the exponential growth of the number of message interchanges and the size of log files. The monitoring and management infrastructure should be either able to handle growing amounts of work in a graceful manner or to be readily enlarged to cope with new workload on the fly (i.e. should be elastic).



**Fig. 1.** Overall SOA4All Architecture

Fig. 1 provides a high-level view on the overall SOA4All architecture centered on the infrastructural components, the main artifacts manipulated and how the SOA4All Studio is integrated. Integration in SOA4All takes place through the so-called Distributed Service Bus (DSB) which integrates notions from Enterprise Service Buses and Semantic Spaces into a unified distributed infrastructure providing message-based and event-based communication, as well as a distributed RDF repository. From a client perspective, the DSB provides access to the platform services as well as the myriad of services provide by third parties. The services offered through the bus include, traditional third-party WSDL and RESTful services, light-weight Semantic Web Services based on annotations over traditional services, and infrastructural services supporting actions such that the discovery, ranking & selection and execution of services.

In a nutshell, the SOA4All Studio is the gateway for the user. It therefore provides a Web-based interface for creating or enhancing annotations, browsing them, discovering suitable services, invoking services, and finally analyzing their execution. These different functionalities are provided by three different platforms composing the Studio. In particular, the Service Provisioning Platform supports the user in providing annotations may them be WSMO-Lite, MicroWSMO, tags or ratings. The Service Consumption Platform allows the user to browse, discover and invoke existing services. Finally the Service Analysis Platform provides the means for users to analyze the execution of services either at runtime or post-execution.

The SOA4All Studio, like every other component is integrated into the overall architecture through the Distributed Service Bus. It is worth noting however that as opposed to platform services that provide infrastructural services, the Studio is mainly a client for these different components, allowing the user to interact with them in a seamless and transparent way. To support this, there is an internal component within the Studio in charge of supporting the interaction with the bus by making use of the appropriate messages and protocols based on Web and WS-\* standards.

The SOA4All Analysis Platform (AP) provides a wide range of widgets in its graphical views, organized according to their potential use and corresponding to the different levels of abstraction [1] that are useful for a variety of stakeholders. Furthermore, the AP offers a completely customizable approach to data visualization so as to correspond precisely to the expectations and needs that more advanced users have. These capabilities are presented in detail in the paper.

## 2 Related Work

Analysis of Provenance information in environments where high volumes of linked-data is combined and consumed is of particular interest in order to evaluate and assess the quality of the data [11].

Methods for representing and analyzing Provenance information have been widely addressed for a long time now [13]. In [14], the authors provide a taxonomy of provenance characteristics, differentiating between data-oriented approaches, focused on data items, and process-oriented approaches, where the emphasis is placed on the processes that generate and consume the data.

We also highlight the Open Provenance Model (OPM, [12]), a community-driven data model for Provenance designed to support interoperability of provenance technology. In OPM, three different pieces of provenance information are identified: artifacts, processes, and agents; while Provenance is represented by graphs, in which the nodes are those elements.

Another approach regarding data provenance is the one by Bunemann et al. [2] where they raise questions such as how the provenance information is obtained, how to cite components of a digital library such as a document in another context, or how to ensure the integrity of citations under the assumption that the cited databases evolve.

Finally, provenance models such as the one proposed by Harth et al. [12] include a “social dimension” which associates provenance information with the originator – which typically will be a person– of such a piece of information. In these models, the social context of the users can be combined with the analysis of provenance information to perform an extra assessment of the quality of the data.

Major efforts have been devoted so far to the monitoring and analysis of processes and services. In [13] the author provides a comprehensive analysis on current techniques in process monitoring and control within organisations. The author covers the topic by addressing four relevant perspectives to process controlling: the data perspective, the usage perspective, the tool perspective and the method perspective. The data perspective is concerned with collecting, storing and representing audit trail information. The author describes existing techniques and proposes an audit trail format that we took into account while devising our conceptualisations. Our approach, although similar in many respects, provides a more formal conceptualisation that is amenable to automated reasoning. This conceptualisation, which underlies the system, provides us with the capacity to apply advanced knowledge-based techniques in a domain-independent manner, as opposed to current practices.

The usage perspective concerns how process controlling management is approached. Some of the state of the art solutions focus on exception handling [14], whereas others focus on the global management of processes [15]. Our work so far has focussed mostly on gathering knowledge about processes in a way that can support automated reasoning as well as it can help business analysts in the management of processes. Although, automated process control is so far not supported by the tool, the very goal of our extensive conceptualisation work has precisely been carried out in order to better support machines in automatically controlling processes [16]. The work carried out so far represents substantial steps in this direction.

The tool perspective is concerned with the architecture and tools that have been developed for process monitoring and control so far. Among these the most relevant to us are for instance the work on PISA (Process Information System based on Access) which precedes the work presented in [13], and the work on the Business Process Intelligence tool suite [17-19]. While the former does not make any use of semantic technologies to enhance the monitoring capabilities the latter uses lightweight taxonomies. Their work is however more focussed on the integration of mining techniques and support for explanations. In this respect we believe both approaches are complementary and consider that a more extensive use of knowledge-based techniques could indeed enhance their results. In a similar way, future work on SENTINEL will indeed be inspired by this research.

On a broader sense, BAM functionality as part of the BPM system is already supported in several products, e.g. [20,21]. Typically the BAM solution is tightly integrated with the BPEL engine which is part of the BPM system. Metrics are defined and calculated based on the events which are published by that BPEL engine. The event publishing mechanism is thereby proprietary. Another approach is to extend the BPEL process with event publishing activities, which invoke operations on a monitoring tool. This approach is utilized in [22] and [23]. The benefit of this approach is that event publishing does not depend on a proprietary mechanism of a BPEL engine. The main disadvantage however is that the BPEL process is more difficult to read and maintain, as it contains new activities, which deal with technical issues, and not just business logic. Our work relies on event publishing mechanism by the execution engines (e.g., BPEL engine). The difference to the existing tools is that in our approach ontologies are used for the description of events and the data they contain. In this way we better support the integration of proprietary formats via ontological mappings, and still allow inferring implicit knowledge at runtime.

### 3 Overall Architecture

This section gives a high-level overview of the Analysis Platform architecture. It is illustrated in Fig. 2., which presents the main components of the platform and their connections to the external components. The elements within the light grey rectangle correspond to the Analysis Platform components and their Studio counterparts, while the elements drawn outside the rectangle represent external components (the Analysis Warehouse and the DSB). The thick arrows illustrate interactions between the Analysis Platform and the external components. The figure also illustrates that the Analysis Platform provides RESTful services that wrap parts of its functionality for easy external use.

The main source of data for the Analysis Platform is the *SOA4All Distributed Service Bus* (DSB), which is the backbone of the SOA4All runtime infrastructure. Several data collectors (bus, grid and engine collectors) available through the bus will feed relevant monitoring data to the AP, which in turn will be able to control and filter data collection through appropriate management operations. In addition, the AP will be able to send management commands to the DSB to instruct its components to perform a variety of infrastructural operations related to the customization or control of DSB components.

We distinguish between two types of monitoring events that can be received from the DSB: *infrastructure events* and *application events*. The former correspond to low-level details of the execution platform while the latter correspond to data about the actual execution of the application services and processes.

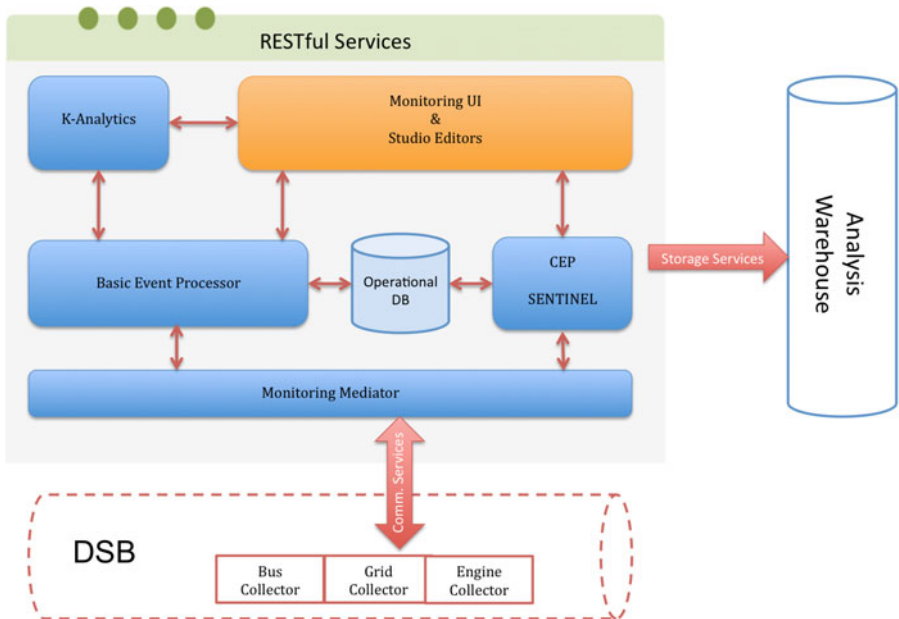


Fig. 2. Overview of the Analysis Platform Architecture

The following are the components of the Analysis Platform.

- **Monitoring Mediator (MM):** Obtains Data from the DSB and the execution engines using the Studio APIs. It is both a listener and a proactive entity: its main role is to interface with the infrastructure (it receives events and can also filter and control the event sources); it can also communicate to the management/monitoring APIs to coordinate event production.
- **Basic Event Processor (BEP):** performs pre-processing of monitoring events from the MM, including computing basic averages and statistics; provides data to Knowledge Analytics and SENTINEL for knowledge extraction; feeds data to basic UI widgets; uses the analysis warehouse to store derived information and basic computation results; the BEP acts as a single point of entry to the analysis warehouse thus interfacing with the other internal components of the Analysis Platform as well as with the Studio editors requesting analysis information to display on diagrams.
- **Knowledge Analytics (K-Analytics) component:** this component will be able to interpret the environment provenance information in order to produce domain-oriented interpretations of process executions to increase user understanding of such executions, which will most likely be potentially very large and complex. Since K-Analytics performs post-mortem analysis, it will query the BEP, and other data sources, for monitoring information when necessary.
- **UI Widgets:** basic graphical representation (as seen in mockups, such as average response time for a service, availability etc.);
- **SENTINEL:** this environment provides knowledge-based techniques in order to detect and diagnose process deviations based on monitoring information and informed by context data.

## 4 Mediation and Basic Event Processing

This section briefly presents the mechanisms used by the Analysis Platform to integrate with the Studio editors and the runtime part of SOA4All as well as the event processing functionalities that are going to be offered by the BEP module.

The **Monitoring Mediator (MM)** is the interface of the Analysis module to the runtime platforms. It connects to the aggregated runtime data collectors and transforms the events received into API calls to the Basic Event Processor. In addition, the Monitoring Mediator is used to relay management operations originating in the Analysis Platform through to the runtime environments.

The Monitoring Mediator leverages mechanisms complying with the WSDM standard to communicate to the data collectors.

The **Basic Event Processor (BEP)** is responsible for

- Parsing the monitoring events received from the MM and to perform data processing in order to extract derived information (such as computing averages and basic aggregated events)

- Communicating raw and derived to upper-level processing entities, the K-Analytics and SENTINEL engines.
- Updating the graphical data structures used by the UI widgets.
- Storing derived events into the analysis warehouse using the Studio Storage Services. These derived events can originate in the BEP itself or indeed can arrive from K-Analytics or SENTINEL.

## 5 SENTINEL

Semantic Business Process Analysis, that is, the extension of Business Process Analysis with Semantic Web and Semantic Web Services technologies has been proposed as a means for increasing the automation of tasks through the provisioning of semantic descriptions of the artefacts involved in the life-cycle of business processes [16]. Research in this direction has already produced an extensive set of ontologies and tools within an overall framework that spans from low-level technical details to high-level methods and tools for analyzing business processes and services to achieve the desired business goals [6, 7, 16, 24]. The tool presented in this paper is part of this overall research thread and as such is strongly based on the use of semantic technologies to enhance the state of the art in Business Process Analysis.

SENTINEL [10] makes use of semantic technologies to facilitate the monitoring of processes and activities. In a nutshell, it uses ontological models for keeping track of execution of processes, and it uses the overall information gathered over time in order to derive business-level knowledge from the low-level trails. In particular, it uses the COBRA and EVO ontologies.

COBRA[9] provides a core terminology for supporting Business Process Analysis (BPA) where analysts can map knowledge about some particular domain of interest in order to carry out their analyses. It is worth noting that COBRA does not aim to provide a fully-comprehensive conceptualisation for supporting each and every kind of analysis since the scope would simply be too big to be tackled appropriately in one ontology. Instead COBRA, depicted in [9] provides a pluggable framework based on the core conceptualisations required for supporting BPA and defines the appropriate hooks for further extensions in order to cope with the wide-range of aspects involved in analysing business processes.

COBRA builds upon Time Ontology that provides a temporal reference by means of which one can determine temporal relations between elements. COBRA provides a lightweight foundational basis in order to ensure a coherent view among additional extensions. It defines concepts such as Process, Activity, Process Instance, Role, Resource, Organisation or Person which are to be refined within specific ontologies as defined within SUPER, or other approaches like the

Enterprise Ontology [7] or TOVE [8]. COBRA has been extended with a reference Events Ontology (EVO) [9] that provides a set of definitions suitable to capture monitoring logs from a large variety of systems and ready to be integrated within our core ontology for analysing business processes.

In order to carry out these activities, SENTINEL makes use of an event processing engine, namely Drools Fusion<sup>2</sup>. Drools provides support for Complex Event Processing over the stream of events generated by the monitoring infrastructure during the execution of processes and services. Based on this stream, the engine is able to identify meaningful events within the cloud, correlate them, reason about the temporal relationships, etc.

The current version of the tool for instance contains rules that process the events collected through the BEP in order to:

- Update the information about lifecycle of business processes and activities therefore tracking all the states through which they have been (e.g., suspended, executing, etc) and the time spent on each of these states;
- Track the involvement of agents (i.e., humans and computers) in processes and activities;
- Compute metrics based on user-defined formulae and whenever certain criteria are met, e.g. compute the execution time of business processes when they finish;
- Compare the results obtained with thresholds determining the limit between “normal” situations and extreme ones;
- Support near real-time visualization of computations performed.

Therefore, in a nutshell, the tool performs continuous data warehousing whereby information coming from the monitoring infrastructure is collected, correlated, processed and stored in a convenient high-level format that better supports its subsequent analysis. On the basis of this higher-level model, additional tools perform near real-time computations when certain criteria are met (e.g., process finished), and periodical computations (e.g., daily batch processing) for less time critical activities. The results are all stored within an integrated analysis warehouse which supports querying and reasoning and therefore it allows the seamless retrieval, interpretation, correlation and combination of all analysis results in order to better support the analysis of processes and services execution by both humans and machines.

The current version of the tool performs the majority of these analyses in a somewhat disconnected way from other results. For instance, thresholds are currently defined manually by users. In the future, however, the idea is to benefit from the integrated analysis warehouse that is populated during execution and analysis, may it be post-execution or real-time, in order to benefit from previously calculated results (e.g., average execution time, deviation of the execution time) to directly establish a reasonable threshold. Similarly, more advanced techniques such for the diagnosis of processes, such as Heuristic Classification [16], have been tested but require additional work in order to find an appropriate trade-off between performance, accuracy, sensibility and the level of prediction.

## 6 K-Analytics: Knowledge Analytics Component

The Knowledge Analytics component (K-Analytics) derives information from several sources in order to deal with it more suitably at an upper-level which end-users can

---

<sup>2</sup> <http://jboss.org/drools/drools-fusion.html>



understand more easily. Hence, the motivation of this component comes from the need to abstract from the many sources of data which include a huge number of logs to a conceptual level that aggregates that information into a more condensed and specific version, in order to facilitate end-users the understanding of all this information.

In SOA4All, we do not rely on the BEP mentioned in the previous section as the only source from which we will get raw information, but we will be making use of further data which is semantically linked, in the form of feedback information and logs which do not come from the execution of the services itself, but from the interactions of the users within the platform, such as the services they open or use.

Therefore, even though the component is open enough to cater for the needs of different situations where the data we are interested in comes from other sources, we will focus in this particular case on the following three ones:

- The **BEP**, which will be queried for the necessary information about executions of the services.
- The **Auditing Framework**, which takes care in SOA4All of semantically logging the actions of the users within the platform (e.g., opening a service, invoking a service), which are stored in the Semantic Spaces via the Storage Services of the SOA4All Studio.
- The **Feedback Framework**, which deals with information of ratings, tags and comments made by the users, also stored in RDFs in the Semantic Spaces via the Storage Services, following the Review Schema<sup>3</sup> and the Tag Ontology<sup>4</sup>.

The aforementioned three sources will obviously yield such a quantity of logs from the project that it will be unfeasible to analyze them at run-time retrieving each piece of information we are interested in. On the contrary, this component performs a post-mortem analysis of those sources in order to make the access to the relevant information easier. Therefore, the purpose of our component is to extract and compute the relevant information and store it again in a more suitable and condensed format at a conceptual level, in such a way that we are able to easily access the results of the computations we are interested in, when desired. The knowledge-level information we aim to infer is supported by an ontological layer, mainly based on the following concepts, which, in turn, come from several useful subconcepts, also addressed:

- **Frequency:** How much is a service used. We derive this concept from Visibility frequency (how many times a service is opened) and Invocation frequency (how many times it is actually consumed)
- **Performance:** How well does the service behave. We derive it from Time performance (how long does the execution take) and Reliability performance (if the execution finishes well)
- **User Perception:** What do the users think about the service. Derived from Ratings perception (average of the ratings) and Reviews perception (number of ratings, comments and tags)

The BEP source is obviously important in order to extract the information about invocation frequency and performance, while the Auditing Framework is the source for

---

<sup>3</sup> <http://purl.org/stuff/rev#>

<sup>4</sup> <http://www.holygoat.co.uk/owl/redwood/0.1/tags/>

the visibility frequency, and the Feedback Framework is the source for the derivation of the User Perception. In turn, the results of the analysis are supported by a generic ontology, based on the aforementioned concepts, which may be combined with suitable domain ontologies for establishing better explanations of the characteristics of each service. Regarding the way K-Analytics operates, in order to extract the upper-level information that we are interested in, our component permits selecting the services we are interested in “tracking”. Hence, when a *track* has been set on a service, our component begins to perform its calculations at batch-time nightly, condensing the great amount of logs that come from different sources into a more appropriate format that the GUI will be able to access more efficiently.

K-Analytics places a special emphasis on the analysis of provenance information, which is a topic of particular interest in a Web of Data where so much interlinked information is used [11], as it actually happens in SOA4All, where a huge number of services are foreseen to be invoked. In particular, we are interested in the so-called *why-provenance* (the origins that were involved in calculating a single entry of a query result), which Buneman et al. [2] distinguished from *where-provenance* (the exact location from where an element of a query has been extracted from), and which is also different from *how-provenance* (the way the origins were involved in the calculation) addressed by Green et al. [3]. Hence, K-Analytics will permit to go from the computed concepts to the actual logs that have generated them, passing through the intermediate sub-concepts that are used for these calculations, and in this way be able to understand the characteristics of the results more deeply when desired.

### 7 Scalability Aspects

This section briefly discusses scalability aspects that are being addressed in the presented Analysis Platform. The discussion is based on a simple example illustrated in Fig 3.

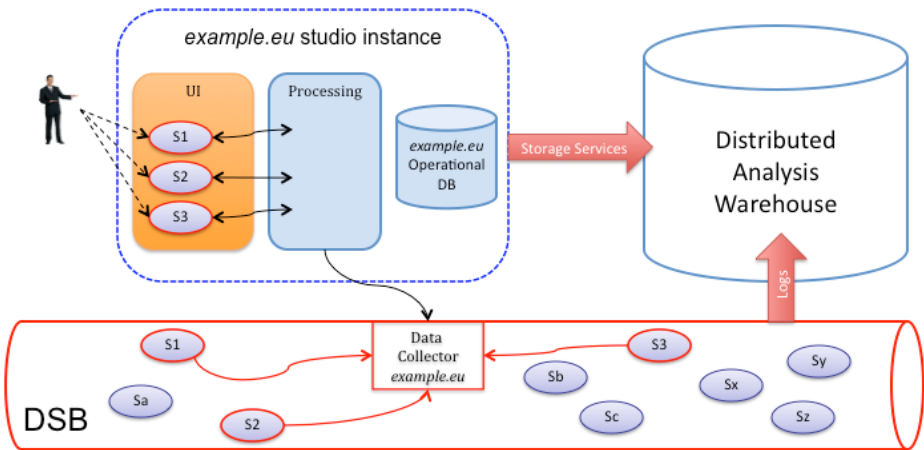


Fig. 3. Illustration of Scalability Aspects

As suggested in the Introduction, in the scenarios envisaged by SOA4All, the proposed Analysis Platform, together with other Studio components, will potentially be deployed on many nodes, each administered by an organisation that wants to provide SOA4All functionality to its users. However, the invisible SOA4All infrastructure based on the distributed service bus can be seen as transcending individual providers and offering a pervasive infrastructure base that is accessible from any SOA4All-enabled node. This is also true for the RDF storage support available as a pervasive service for SOA4All nodes. This pervasive aspect is realised through different federation techniques that are out of the scope of this paper.

This architecture however has important implications with regard to scalability of the Analysis Platform. In order to cope with the extremely large numbers of users envisaged by SOA4All as a whole, as well as potentially massive amounts of data, we leverage the natural distribution provided by the infrastructure and optimize data processing and storage to achieve a good compromise in terms of performance and scalability. Fig 3 shows a sample instance of the Studio called *example.eu*. This Studio instance runs its own individual Analysis Platform instance, separated from other instances available with other providers. Users connecting to *example.eu*'s Studio will access this version of the Analysis Platform and will be interested, as a group, in a relatively limited number of services and processes (that are relevant to *example.eu*'s business). This implies that all the processing that this instance of the Analysis Platform will perform will relate to these services (illustrated in the image as S1, S2 and S3). Its internal operational database will contain detailed analysis data for these items. This prevents this instance from processing data for services that are not of interest to its users (in effect, only services that at least one user of the domain is interested in, will be analyzed).

Naturally, *example.eu*, as all the other SOA4All-enabled nodes running the Studio, will connect to the DSB and will leverage the highly-distributed RDF storage. This enables the collection of data for any service or process executing anywhere in the world, when such data is required. We envisage that even for services/processes that no user has expressed interest in yet (in any domain instance), the DSB will collect basic data such as moving averages for execution times and availability information, and store it in the RDF storage, in order to have minimum bootstrap information ready when users become interested in the particular service/process. As soon as they do become interested, data collection becomes much more significant as it is driven by the Analysis Platform instance, and detailed analysis data can be stored in the individual operational databases. In short, detailed analysis for selected entities is performed "locally" in the same domain as the user, and basic analysis and long-term storage is performed on the distributed infrastructure for all entities.

## 8 Information Presentation

The widgets, consoles and screens used by the Analysis Platform are accessible in a unified and integrated manner through the SOA4All Studio.

Monitoring information is often structured around three different views [4]: (i) the Process View which is concerned with key performance indicators of processes and services; (ii) the Resource View centered around the resources, human or mechanized,

required for executing processes; and (iii) the Object View which focuses on business objects such as inquiries, orders or claims. These three views are populated with statistical information such as the minimum, the maximum, the average or the deviation of some parameter of interest.

These views are of major relevance to analysis and management, and as a consequence they are typically supported by monitoring tools such as Business Activity Monitoring solutions. However, different users have different roles, interests, access rights, and preferences and these vary depending on the specific scenario, the focus of their analysis, etc. The user interface of a fully-fledged general purpose solution must therefore be characterized by its flexibility [4]. This includes for instance support querying, filtering and ordering the information according to user defined specifications [5]. Indeed, given the kinds of users addressed, the specification of these queries and filters should be supported in a simple way so that humans can browse the existing execution information effectively. Similarly, different domains exhibit particular characteristics, which impede a “one size fits all” approach. The monitoring tool should therefore support users in defining their own visualization templates to be populated with relevant monitoring information. The visualization framework should be supported by a wide range of graphical representations such bar charts, line charts, pie charts, time series charts, etc. Additionally, the visualisation framework should support the presentation of user-defined information combining diverse statistical information about processes, etc.

The use of knowledge-based technologies will play a major role in bringing flexibility to the Analysis Platform to be able to adapt to a myriad of users and services in a seamless way. On the one hand the use of a formal conceptual model closer to human understanding than low-level syntactic representations will bring the body of knowledge to a higher level of abstraction more suitable for human interpretation. On the basis of this conceptual model we shall support humans in defining queries or navigating through the data by simply following the conceptual schema and generating the appropriate ontological queries transparently. This will allow, among other things, to seamlessly navigate across different abstraction layers. Additionally, we envisage the use high-level strategic models such as the one presented in [6] in order to guide the presentation of analysis information driven by the importance and impact data can have on underlying services and their related interdependencies.

## 9 Conclusion and Future Work

Large-scale web-based service platforms require the support of comprehensive monitoring and analysis tools if they are to become widely used. This paper presented our approach for an Analysis Platform that we believe is suitable in such contexts. A detailed description of the current prototype, which is in an advanced stage, can be found in [28].

There are however important remaining developments for the AP. Different editors that will likely be used by end-users to create processes and services need to use information generated by the AP to augment their graphical displays. The APIs that are currently available may need to be refined to correspond to this need.

The AP needs to better leverage semantic information about services and processes and update semantic repositories with more analysis and monitoring information. This

information could in future be used by ranking and selection mechanisms that aid users in finding the best services for their needs.

One of the main remaining challenges however refers to the scalability ambitions of the envisaged context, which may involve millions of services used by numerous users. The current prototype already supports a distributed environment and it will need to be tested and potentially refactored to correspond to the wide distribution of its data sources both from functional as well as non-functional points of view (i.e. dealing with large amounts of data in information presentation as well as ensuring that the AP infrastructure can cope with the demands). We are confident that the decoupled, distributed approach we have taken will enable us to reach this goal and we look forward to the validation tests that will be carried out through SOA4All.

## Acknowledgements

The authors wish to gratefully acknowledge the support for this work provided by the European Commission through the SOA4All project.

## References

1. Mos, A., Boulze, A., Quaireau, S., Meynier, C.: Multi-layer perspectives and spaces in SOA. In: Proceedings of the 2nd international Workshop on Systems Development in SOA Environments (SDSOA 2008), Leipzig, Germany (May 2008)
2. Buneman, P., Khanna, S., Tan, W.C.: Why and Where: A Characterization of Data Provenance. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, p. 316. Springer, Heidelberg (2000)
3. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance Semirings. In: Proceedings of the 26th Symposium on Principles of Database Systems (PODS). ACM, New York (June 2007)
4. Muhlen, M.z.: Workflow-based Process Controlling. In: Foundation, Design, and Implementation of Workflow-driven Process Information Systems, Logos, Berlin, vol. 6 (2004)
5. Hur, W., Bae, H., Kang, S.-H.: Customizable Workflow Monitoring. *Concurrent Engineering* 11(4), 313–325 (2003)
6. Pedrinaci, C., Markovic, I., Hasibether, F., Domingue, J.: Strategy-driven business process analysis. In: 12th Conference on Business Information Systems, BIS (2009)
7. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The Enterprise Ontology. *Knowledge Engineering Review* 13(1), 31–89 (1998)
8. Fox, M.S.: The TOVE Project Towards a Common-Sense Model of the Enterprise. In: Belli, F., Radermacher, F.J. (eds.) IEA/AIE 1992. LNCS, vol. 604, pp. 25–34. Springer, Heidelberg (1992)
9. Pedrinaci, C., Domingue, J., Alves de Medeiros, A.K.: A Core Ontology for Business Process Analysis. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 49–64. Springer, Heidelberg (2008)
10. Pedrinaci, C., Lambert, D., Wetzstein, B., van Lessen, T., Cekov, L., Dimitrov, M.: SENTINEL: A Semantic Business Process Monitoring Tool. In: Workshop: Ontology-supported Business Intelligence (OBI 2008) at 7th International Semantic Web Conference (ISWC 2008), Karlsruhe, Germany (2008)
11. Hartig, O.: Provenance Information in the Web of Data. In: Proc. of the Linked Data on the Web Workshop at WWW (2009)

12. Moreau, L., Plale, B., Miles, S., Goble, C., Missier, P., Barga, R., Simmhan, Y., Futrelle, J., McGrath, R., Myers, J., Paulson, P., Bowers, S., Ludaescher, B., Kwasnikowska, N., Van den Bussche, J., Ellkvist, T., Freire, J., Groth, P.: The Open Provenance Model. Technical report, Electronics and Computer Science. University of Southampton (2008)
13. Bose, R., Frew, J.: Lineage retrieval for scientific data processing: A survey. *ACM Computing Surveys* 37(1), 1–28 (2005)
14. Simmhan, Y., Plale, B., Gannon, D.: A Survey of Data Provenance in e-Science. *SIGMOD Record* 34(3), 31–36 (2005)
15. Harth, A., Polleres, A., Decker, S.: Towards a Social Provenance Model for the Web. In: *Proceedings of the Workshop on Principles of Provenance* (November 2007)
16. Mühlen, M.: Workflow-based Process Controlling. In: *Foundation, Design, and Implementation of Workflow-driven Process Information Systems. Advances in Information Systems and Management Science*, vol. 6. Logos, Berlin (2004)
17. Casati, F., Castano, S., Fugini, M., Mirbel, I., Pernici, B.: Using patterns to design rules in workflows. *IEEE Trans. Softw. Eng.* 26(8), 760–785 (2000)
18. Kueng, P., Kueng, P., Meier, A., Meier, A., Wettstein, T.: Computer-based Performance Measurement in SME's: Is there any option. In: *Proceedings of the International Conference on Systems Thinking in Management*, pp. 8–10 (2000)
19. Alves de Medeiros, A.K., Pedrinaci, C., van der Aalst, W., Domingue, J., Song, M., Rozinat, A., Norton, B., Cabral, L.: An Outlook on Semantic Business Process Mining and Monitoring. In: *Proceedings of International IFIP Workshop On Semantic Web & Web Semantics, SWWS 2007* (2007)
20. Sayal, M., Casati, F., Dayal, U., Shan, M.-C.: Business process cockpit. In: *VLDB 2002: Proceedings of the 28th international conference on Very Large Data Bases. VLDB Endowment*, pp. 880–883 (2002)
21. Grigori, D., Casati, F., Castellanos, M., Dayal, U., Sayal, M., Shan, M.-C.: Business Process Intelligence. *Computers in Industry* 53(3), 321–343 (2004)
22. Castellanos, M., Casati, F., Dayal, U., Shan, M.-C.: A comprehensive and automated approach to intelligent business processes execution analysis. *Distributed and Parallel Databases* 16(3), 239–273 (2004)
23. Wahli, U., Avula, V., Macleod, H., Saeed, M., Vinther, A.: *Business Process Management: Modeling through Monitoring Using WebSphere V6.0.2 Products*. Number SG24714801. IBM RedBooks (2007)
24. Oracle Corporation. Oracle Business Activity Monitoring (BAM) (2008), <http://www.oracle.com/appserver/business-activity-monitoring.html>
25. Baresi, L., Ghezzi, C., Guinea, S.: Smart monitors for composed services. In: *ICSOC 2004: Proceedings of the 2nd international conference on Service oriented computing*, pp. 193–202. ACM Press, New York (2004)
26. Roth, H., Schiefer, J., Schatten, A.: Probing and Monitoring of WSBPEL Processes with Web Services. In: *CEC-EEE 2006: Proceedings of the The 8th IEEE International Conference on E-Commerce Technology and The 3rd IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services*, Washington, DC, USA, p. 30. IEEE Computer Society, Los Alamitos (2006)
27. Alves de Medeiros, A.K., Van der Aalst, W., Pedrinaci, C.: Semantic Process Mining Tools: Core Building Blocks. In: *16th European Conference on Information Systems*, Galway, Ireland (June 2008)
28. Mos, A., Pedrinaci, C., Rey, G.A., Martinez, I., Hamerling, C., Vaudaux-Ruth, G., Liu, D., Quaireau, S.: D2.3.2 Service Monitoring and Management Tool Suite First Prototype, SOA4All Deliverables, <http://www.soa4all.eu/file-upload.html?func=startdown&id=129>