Semantic Technologies for Enterprise Cloud Management

Peter Haase, Tobias Mathäß, Michael Schmidt,
Andreas Eberhart, Ulrich Walther

fluid Operations, D-69190 Walldorf, Germany
`firstname.lastname@fluidops.com`

Abstract. Enterprise clouds apply the paradigm of cloud computing to enterprise IT infrastructures, with the goal of providing easy, flexible, and scalable access to both computing resources and IT services. Realizing the vision of the fully automated enterprise cloud involves addressing a range of technological challenges. In this paper, we focus on the challenges related to intelligent information management in enterprise clouds and discuss how semantic technologies can help to fulfill them. In particular, we address the topics of *data integration*, *collaborative documentation and annotation* and *intelligent information access and analytics* and present solutions that are implemented in the newest addition to our eCloudManager product suite: The Intelligence Edition.

1 Introduction

Cloud computing has emerged as a model in support of "everything-as-a-service" (XaaS) [10]. Cloud services have three distinct characteristics that differentiate them from traditional hosting. First, cloud services are sold on demand, typically by the minute or the hour; second, they are elastic – users can have as much or as little of a service as they want at any given time; and third, cloud services are fully managed by the provider (while the consumer needs nothing but a personal computer and Internet access) [14]. Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet and a weak economy, have accelerated interest in cloud computing.

While the paradigm of cloud computing is best known from so called public clouds, its promises have also caused significant interest in the context of running enterprise IT infrastructures as private clouds [12]. A private cloud is a network or a data center that supplies hosted services to a limited number of people, e.g. as an enterprise cloud. As with public clouds, the goal of enterprise clouds is to provide easy, scalable access to computing resources and IT services [15].

The emergence of cloud offerings such as Amazon AWS or Salesforce.com demonstrates that the vision of a fully automated data center is feasible. Recent advances in the area of virtualization make it possible to deploy servers, activate network links, and allocate disk space virtually via an API rather than having to employ administrators who physically carry out these jobs. Note that virtualization is not limited to CPU virtualization – virtualization can be defined as an abstraction layer between a consumer and a resource that allows the resource to be used in a more flexible way. Examples can be drawn from the entire IT stack. Storage Area Networks (SANs) virtualize mass storage resources, VLAN technology allows using a single physical cable for multiple logical networks, hypervisors can run virtual machines by presenting a virtual hardware interface to the guest operating system, and remote desktop software such as VNC virtualizes the screen display by redrawing it on a remote display.

Realizing the vision of the fully automated data center – the enterprise cloud – involves addressing a range of technological challenges, touching the areas of infrastructure management, virtualization technologies, but also distributed and service-oriented computing. In this paper, we focus on the challenges related to intelligent information management in enterprise clouds and discuss how semantic technologies can help to fulfill them. In particular, we address the topics of *data integration*, *documentation and annotation*, and *intelligent information access and analytics*. We present solutions that we have implemented in the newest addition to our eCloudManager product suite: The Intelligence Edition. In the following, we give a brief overview of the contributions in each of the dimensions.

Data Integration Clearly, being able to automate data center operations via low level APIs is the prerequisite for achieving the requirements listed above. The challenge lies in the proper integration of data received from infrastructure components and the orchestration of subsequent actions as a response to events such as user requests or alarms. Many layers play a role in this picture and one is faced with a large set of provider APIs ranging from storage to application levels. The situation grows even more complex when products and solutions from different vendors coexist in the data center. In fact, CIOs tend to mix hardware from different vendors to avoid vendor locks, in order to benefit from price competitions among the individual vendors. Hence, in the end they often face a mix of technologies acquired over several years, where products from different vendors and sometimes even different product versions differ vastly in syntax and semantics of the data supplied and functionality offered via APIs.

Semantic technologies have been designed for these real-world situations. In our solution, we use RDF as a data model for integrating semantically heterogeneous information sources in order to get a complete picture across the entire data center, both horizontally – across different product versions and vendors – and vertically – across storage, compute units, network, operating systems, and applications. The RDF-based integration offers the flexibility needed to integrate new sources in the presence of heterogeneity and dynamics in data centers.

Collaborative Documentation and Annotation Data integration is a key aspect of running data centers and clouds in an efficient and cost effective way. For this purpose, cloud management software is fed with data from provider APIs. This data contains technical information about the infrastructure and the software running on it. In order to have a complete picture available, organizational and business aspects need to be added to the technical data. Consider the following examples: The decision whether to place a workload on a redundant cluster with highly available storage is strongly affected by the service level the system needs to meet, data center planning tools must take expiring warranties of components into account, and having a relatively mild punishment for SLA violations may lead a cloud operator to take a chance and place workloads on less reliable infrastructure.

In order to collaborate efficiently, data center operators need to document procedures and log activities. Proper knowledge management is essential in order to avoid a problem having to be resolved repeatedly by different staff members. Activities are usually managed via a ticketing system, where infrastructure alerts and customer complaints are distributed and resolved by operators.

The examples above show that business and organizational information must be addressed in a unified way. When information about systems or customers is stored or documentation about a certain hardware type is written, it should be possible to cross reference information collected from infrastructure providers. In our solution we apply Semantic Wiki technology to satisfy these requirements. Operating on an RDF base that is fed by infrastructure providers, operators can extend this data by documenting and annotating the respective items.

Intelligent Information Access and Analytics Efficient management of a data center requires providing data center managers with the information they need to make intelligent, timely and precise decisions. The range of specific information needs that should be supported is very diverse, including the generation of reports about status and utilization of data center resources over time, the visualization of key performance metrics in dashboards, the search for specific resources etc. Many of these information needs require multi-dimensional queries that span across both technical, IT-related aspects and business aspects, and therefore cannot be answered by a single data source alone.

Enabled by an integrated view on the data, we support queries that overcome the borders of data sources. Apart from predefined queries that drive reports and dashboards, a clear benefit is also the ability to perform expressive *ad hoc* queries. As it is desirable to hide the complexity of the underlying data model and query languages from the end user, we use approaches of schema-agnostic semantic search that combine the expressiveness of structured queries with the ease of use of keyword driven interfaces. Novel approaches to the visualization of structured data as well as visual exploration of resources enable new forms of interaction with the resources and provide insights into previously hidden relationships.

Structure of the Work We start with a brief overview of the overall solution in Section 2. Next, Section 3 introduces our ontology for the domain of enterprise cloud management, which serves as the core of the data integration and management tasks within the eCloudManager. We then discuss the specific uses of semantic technologies: semantics-based integration in Section 4, collaboration support in Section 5 and intelligent information access in Section 6. Subsequently, we report on practical experiences in Section 7 and, after a discussion of related work in Section 8, conclude with an outlook to future work in Section 9.

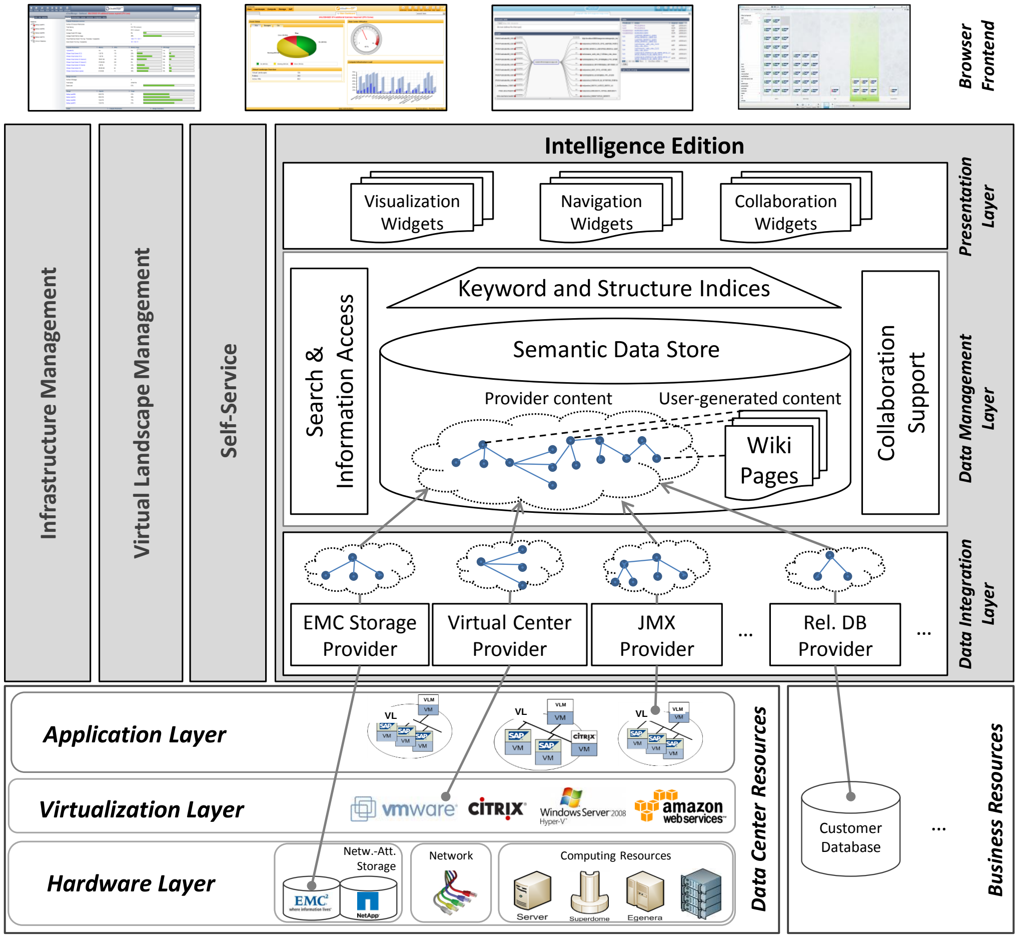


Fig. 1. eCloudManager Architecture

2 Solution Overview

The eCloudManager Product Suite is a Java-based software solution that is targeted at the management of enterprise cloud environments. Starting with the big picture, we first discuss the eCloudManager's overall architecture depicted in Figure 1. The bottom of the figure shows the two dimensions of information relevant to the eCloudManager, namely *Data Center Resources* and *Business Resources*. The data center resources are divided along the IT stack into (i) a *Hardware Layer* that consists of physical storage, network and compute infrastructure, (ii) a *Virtualization Layer* built on top of the hardware layer that is made up of hypervisors with appropriate management capabilities (enabling virtual clusters, live migration etc.), and finally (iii) the *Application Layer* built on top of the virtualization layer, comprising applications and landscapes on top of the virtualized resources (where the term *landscape* refers to a set of enterprise systems plus optional auxiliary systems that enable network access using VPN, RDP, SAP GUI or other means). These data center resources are complemented by associated business resources, like customer data, hardware catalogs including component descriptions and pricing data, or related project information.

Built on top of this resource infrastructure, the eCloudManager comes with four complementary editions. Although the focus of this paper is on the *Intelligence Edition* (the large box on the center right of Figure 1), for completeness we shortly sketch the three remaining editions first (the three boxes on the center left). The leftmost edition, *Infrastructure Management*, provides solutions for monitoring and managing tasks ranging over the whole IT stack, like CPU and storage virtualization across different virtualization and storage providers, through a unified graphical interface. Its central features are rapid VM provisioning utilizing multi-vendor server virtualization and storage technologies, as well as error detection through a customizable event and notification system. Complementarily, the *Virtual Landscape Management* edition implements the novel idea of Landscape-as-a-service (LaaS), i.e. to offer up interconnected multitiered multi-system enterprise applications (like complex SAP landscapes) as complete and ready-to-run landscapes. While the Virtual Landscape Management edition is rather administrator-focused, the *Self-Service* edition constitutes an end-user oriented portal for template-based on-demand provisioning of application landscapes for development, value prototyping, testing and production. A unique feature of this edition is a module for metering and billing.

In the remainder of this paper, we will focus on the features of the fourth edition, namely the *Intelligence Edition*, which makes use of innovative semantic technologies to integrate available resources into a semantic data store, investigate this data, and collaboratively interact with the integrated data.

At the bottom of the Intelligence Edition is the *Data Integration Layer*, which relies on the concept of so-called *data providers*. Abstracting from the details, a data provider is a component that extracts data from a single physical or logical resource (e.g. an EMC storage device, a VMware Virtual Center, or a relational database), converts it into RDF and integrates the resulting RDF data into the central repository. It is crucial that several providers can – and typically do – coexist and that data from different providers is automatically interlinked by use of common URIs, ultimately providing a unified and integrated view on all data center and business resources (for more details see Section 4).

The central repository where the provider data is written to is settled in the *Data Management Layer*. Technically, it is realized as a Sesame [6] triple store that adheres to a predefined (yet extendable) OWL ontology (cf. Section 3). In addition to the repository, the layer provides components for search and intelligent, semantics-based information access (cf. Section 6), whose efficiency is made possible by keyword and structure indices over the data. A central component in this layer are also semantic wiki pages that are associated with the resources contained in the repository; they offer an entry point to the eCloudManager users, allowing to add new and complement existing information (cf. Section 5).

The uppermost layer in the Intelligence Edition is the *Presentation Layer*. Located on top of the Data Management Layer, it comes with a predefined set of widgets with varying functional focus, e.g. offering support to display wiki pages, visualize the underlying data using charts and diagrams, navigate through the underlying RDF graph, and collaboratively annotate resources in the database using both semantic annotations as well as free-text documentation. Ultimately, the combination of these widgets results in a customizable user-interface – realized as an Ajax-based browser frontend – which is flexibly adjustable to fit the needs of different user types (like data center administrators or CIOs).

3 The eCloudManager Ontology

Having presented the overall architecture, we now take a closer look at eCloudManager data model and introduce a conceptual model for the domain of enterprise cloud management in form of an ontology that abstracts from vendor-specific representations, data sources and management APIs. The ontology has been modeled as an OWL 2 ontology, consisting primarily of a class hierarchy and property definitions with domain, range and cardinality restrictions. While in the current application many of the expressive features of OWL 2 are not yet required, we opted for OWL 2 in order for the ontology to serve as a reference ontology for the data center domain and to perspectively enable reasoning-driven tasks in the application (cf. the discussion of future work in Section 9).

Figure 2 surveys the main concepts and relationships of the eCloudManager ontology (using the UML profile for modeling OWL ontologies [5]). Each concept carries a number of data properties (attributes), which capture information about properties and status of the respective resource; for space limitations, we included only the most important subsets of attributes. We next describe the four major subareas of the figure in more detail, namely *Storage Infrastructure*, *Compute Infrastructure*, *Application-level* and *Business-level* resources.

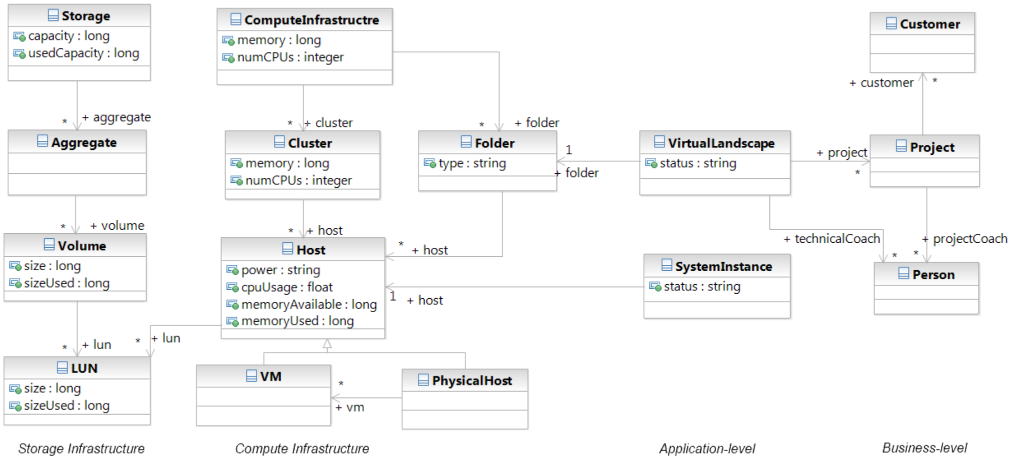


Fig. 2. Overview of the eCloudManager Ontology

Infrastructure Resources Infrastructure resources include storage and compute resources. In our enterprise cloud solutions, storage virtualization is realized through Network Attached Storage (NAS), in which virtual disks are managed as so-called `LUN`s. This allows to maintain complete systems in a central place (the `Storage` filer) and dynamically relocate between hosts. Within a `Storage` filer, `LUN`s are typically grouped and managed in `Volume`s as logical container, which can further be grouped in `Aggregate`s.

Specific solutions for compute virtualization environments are modeled as `ComputeInfrastructure`, representing a certain technology such as FlexFrame, AWS, XEN or VMware. Within an `ComputeInfrastructure`, a `Cluster` represents a collection of compute resources, specifically a group of tightly coupled computers that work together closely and in many respects can be viewed as though they are a single resource. A `Folder` is a logical set of resources that is grouped together, assigned resources, managed etc. Examples are VMware resource pools. A `Host` represents an actual – physical or virtual – compute system (`PhysicalHost` or `VM`), where one `PhysicalHost` can host multiple `VM`s. As shown in the figure, the primary connection between compute and storage resources is given by the `lun` relationship between `Host`s and the logical storage units (`LUN`s) they are associated with.

Application-level Resources In a virtual environment, complex application landscapes can be managed as a `VirtualLandscape`, which is assigned a `Pool` of virtualized infrastructure resources. A virtual landscape typically comprises a set of applications that run as `SystemInstance`s on dedicated hosts. Examples of a virtual landscape would be a landscape of SAP system instances as applications, e.g. consisting of a CRM, ERP and enterprise portal.

Business-level Resources In addition to the technical aspects of the data center (which is typically populated automatically from management APIs), the ontology allows to relate data center resources to relevant business information, such as a `Person` responsible for the administration of data center resources, or related project and customer information etc.

To wrap up, the eCloudManager ontology introduces the basic concepts that are commonly used in enterprise cloud management scenarios. As described in Section 5, it can easily be extended by the user to capture information relevant for specific use cases or to integrate other data sources.

4 Semantics-based Integration

As sketched in Section 2, semantic data integration is realized through *data providers*. Recalling the main idea, a data provider extracts data from a data source, converts this data into RDF, and integrates this RDF data into the semantic data repository. As discussed in the previous section, in the context of enterprise cloud management we are primarily interested in physical properties of hardware components like storage devices, available CPUs, physical hosts, as well as the properties and the current state of software components like hypervisors and virtual machines. Addressing these information needs, the eCloudManager comes with a broad set of predefined data providers for integrating data from hard- and software components typically encountered in the context of enterprise cloud management. To illustrate the concept by example, let us consider a small data center employing a single Xen hypervisor that runs a couple of virtual machines over a set of physical hosts and two storage systems, say one EMC Symmetrix and one NetApp system. In this scenario, data integration in the eCloudManager would be realized through the following providers.

- A Symmetrix provider extracts physical properties from the Symmetrix storage unit such as number and size of physical disks contained in the system.
- A NetApp provider analogously extracts the physical properties from the NetApp storage system.
- Complementarily, a Xen provider gathers the properties hidden behind the Xen hypervisor, such as the number of the virtual machines running on top, configuration details of the individual virtual machines, information about underlying physical hosts, disk occupation of the virtual disks, etc.

Technically, the data providers are predefined modules that can be instantiated for existing data sources and, given only meta information like their network address and login information as input, use available drivers or APIs to gather the relevant data. This data is then translated into RDF, thereby using the key attributes of resources to generate URIs that uniquely identify these resources. To give an example, the key for the Symmetrix storage device is its IP address, so its URI might look like `http://my.datacenter/Symmetrix/192.168.0.55`. The crucial thing here is that both the Symmetrix provider and the Xen provider will generate this URI when storing information related to the Symmetrix system. This way, the data that is generated from different providers is automatically interlinked when integrating it into the global repository (cf. Figure 1).

When instantiating the providers, one can also configure advanced properties like the interval for periodically refreshing data, or whether to store old provider snapshots for recovery purpose. Ultimately, the provider framework serves as an abstraction layer for data integration that allows to connect data sources in a plug & play fashion to the central semantic repository, while abstracting from the technical details and the APIs of the physical and logical components accessed by the provider. If the data center is extended by, say, another Symmetrix storage system, all that needs to be done is instantiating a fresh Symmetrix provider that collects its data (which can be done by an administrator using the eCloudManager Web frontend). A second benefit of the provider concept is the high degree of reuse, e.g. the NetApp provider can be employed in every data center using a NetApp storage system. Taken all these properties together, our data provider concept enables a fast and simple data integration process (cf. Section 7).

Apart from the integration of technical data directly related to the enterprise cloud data center, the provider concept also makes it possible to integrate data from other sources, such as additional documentation or customer information (e.g. in the form of spreadsheets or relational databases). To this end, the eCloudManager offers predefined, customizable providers, e.g. to extract RDF data from external sources, SOAP-based Web Services, SNMP- and SSH-connected devices, relational DBs or tabular CSV files. In addition, there are script-based providers that allow users to integrate data from virtually every accessible source. The underlying idea is to support data integration in a pay-as-you-go fashion: while data center information is typically directly aligned with the eCloudManager ontology presented in Section 3, for other data sources it is up to the user how much effort she wants to spend on data integration: In some cases, only little effort is needed and generic, predefined providers can be used to integrate data sources quickly, leaving it to the user to semantically annotate and interlink this data afterward (cf. Section 5); in other cases, the user may put more effort in data integration and write his own provider that aligns the integrated data directly with existing data and the eCloudManager ontology. In the end, this concept gives users a high degree of flexibility in data integration.

5 Collaborative Documentation and Annotation

As motivated in the Introduction, in order to collaborate efficiently data center administrators need to be able to document and share knowledge. Wikis have long been used as a tool for knowledge sharing in administration environments, yet a major obstacle of traditional wikis is that they again are silos, with largely unstructured information that is poorly connected with other data sources. As outlined in Section 2, in our solution we tightly integrate Semantic Wiki functionality, where the wiki pages are directly associated with the structured data objects. Our wiki implementation follows the ideas of the Semantic MediaWiki [9], providing features like semantic annotations or built-in support for visual display of information (e.g. as charts or bar diagrams). Semantic annotations are automatically extracted from the wiki pages, converted into RDF triples and persisted in the repository. In the user interface, the structured and unstructured information then is presented in a seamlessly integrated way. In the following paragraphs, we discuss uses cases around user-driven documentation and annotation, describing how they are supported in our solution.

Documentation Technical documentation is a central use case in data center management. Examples include customized installation instructions or best practices for handling errors on a host system of the cloud that is known to frequently cause problems. Such documentation, typically provided in form of unstructured text (possibly with some internal and external links), can directly be entered in the wiki pages associated with the respective resource in the central repository.

Interlinkage of Resources In addition to unstructured documentation, the user can use semantic annotations to establish new relationships between resources. For example, one may link responsible administrators directly to hardware devices, or associate customers with the concrete virtual landscapes they are using.

Completion of Missing Information In some cases, data providers may not have access to all information relevant to the user, which ultimately leads to incomplete information in the central repository. Our approach here is to use the eCloudManager ontology to identify missing information: whenever displaying a resource of a given type, we compare its properties against the properties specified in the ontology for the respective type, to identify properties that are currently not populated. For instance, the ontology may define that the warranty status of storage resources should be provided, and this property may not have been specified for some storage system. Using structured tabular templates that are automatically generated from the ontology, the user can easily detect properties with missing values and, in response, directly fill in the missing data.

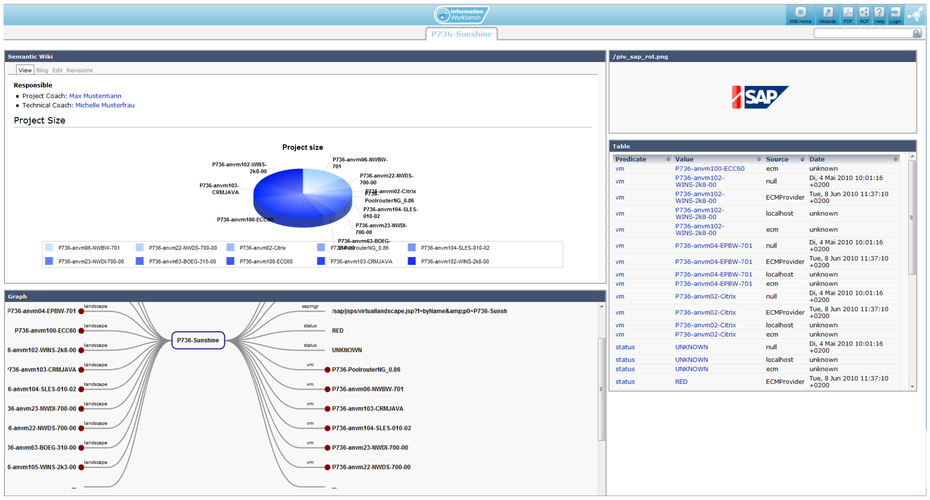
It is worth mentioning that the wiki is equipped with a revision management system to track the provenance of changes. For the structured data, the provenance is maintained down to the level of single triples, i.e. for every information item it is possible to see when and by whom (either a user or a data provider) it was changed. Via RSS feeds users can get notified in the case of changes, e.g. when the status of an object is updated, or a corresponding wiki page has been modified. With these functionalities, the system offers a wide range of possibilities to support collaborative working processes in a data center environment.

6 Semantics-based Information Access

Having discussed the collaborative annotation and documentation features in the previous section, we now present our techniques and paradigms to semantics-based information access. We divide our discussion into three areas, namely *presentation of resources*, *search and querying* and *exploration and analytics*.

Presentation of Resources in the UI The user interface of the Intelligence Edition follows a resource centric presentation scheme, i.e. every resource has exactly one page associated that aggregates resource-related data in a transparent way. The UI is composed from a set of widgets with different functional focus, like visualization, navigation or collaboration. Some of these widgets are generic and displayed for all resources, while others are specific for a fixed type of data (or, alternatively, data with certain properties). To give some concrete examples, the widgets include a browsable graph view, a tabular view to edit structured data, the semantic wiki to create documentation and semantic annotations, as well as widgets displaying charts or bar diagrams. Figure 6 shows an example of a page for a virtual landscape, including a tabular view, the data graph and a dashboard displaying statistics.

Search and Querying As argued in the introduction and solution overview, one particular strength of having data center and business resources integrated in a single repository is the ability to perform queries across the borders of data sources. Following a pay-as-you-go approach, we implement a variety of semantics-based paradigms to searching and querying the integrated data.



Responsible

- Project Coach: Max Mustermann
- Technical Coach: Michelle Musterfrau

Project Size

Table

Predicate	Value	Source	Date
vm	P736-anvm100-ECC60	ecm	unknown
vm	P736-anvm102-WINS-2k8-00	null	Di, 4 Mai 2010 10:01:16 +0200
vm	P736-anvm102-WINS-2k8-00	ECMProvider	Tue, 8 Jun 2010 11:37:10 +0200
vm	P736-anvm102-WINS-2k8-00	localhost	unknown
vm	P736-anvm102-WINS-2k8-00	ecm	unknown
vm	P736-anvm04-EPBW-701	null	Di, 4 Mai 2010 10:01:16 +0200
vm	P736-anvm04-EPBW-701	ECMProvider	Tue, 8 Jun 2010 11:37:10 +0200
vm	P736-anvm04-EPBW-701	localhost	unknown
vm	P736-anvm04-EPBW-701	ecm	unknown
vm	P736-anvm02-Citrix	null	Di, 4 Mai 2010 10:01:16 +0200
vm	P736-anvm02-Citrix	ECMProvider	Tue, 8 Jun 2010 11:37:10 +0200
vm	P736-anvm02-Citrix	localhost	unknown
vm	P736-anvm02-Citrix	ecm	unknown
status	UNKNOWN	null	Di, 4 Mai 2010 10:01:16 +0200
status	UNKNOWN	localhost	unknown
status	UNKNOWN	ecm	unknown
status	RED	ECMProvider	Tue, 8 Jun 2010 11:37:10 +0200

Fig. 3. Screenshot Displaying a Virtual Landscape Resource

Keyword Search The simplest search paradigm from a user's perspective is keyword search. In contrast to classical keyword search, in our scenario search is not document- but entity-centric: When answering a keyword query, we investigate both the data properties and the associated unstructured data (i.e., the wiki pages) for each entity and, if the keywords are matched, return the respective entity. Hence, keyword search is well-suited to quickly locate entities with known properties and/or annotations. Technically, our search engine uses the Lucene[1] library to maintain the keyword indices for RDF literals and wiki pages.

Structured Queries using SPARQL Advanced users may also be interested in answering more complex queries, e.g. asking for all projects of a certain costumer together with billing contact information of this costumer. In such cases, the RDF query language SPARQL[2] can be used to encode requests directly.

While the SPARQL query language is very expressive (i.e., relationally complete [3]), formulating SPARQL queries requires knowledge about the query language and the schema of the underlying data. To equip users with the high precision of structured search without having detailed knowledge about the schema, we support a variety of means to perform ad hoc queries in a schema-agnostic way [13], as dicussed in the following.

Form Based Search We developed a form based search interface that supports the user in formulating SPARQL queries by presenting templates the user has to complete. This way the user can apply the expressiveness of a structured query language without directly being confronted with its concrete syntax. The

---

[1] `http://lucene.apache.org/`

[2] `http://www.w3.org/TR/rdf-sparql-query/`

templates are directly derived from the ontology. For instance, when searching for entities of type "Virtual Landscape", the dynamic form automatically searches for and presents properties having domain `VirtualLandscape`. The user can fill in values for the properties she is interested in, such as e.g. the virtual landscape's status or associated resource pools.

The form based approach is one way to bridge the gap between easy-to-use keyword search and precise structured search. It should be noted that, while greatly facilitating the access to precise structured search, form based search also restricts the expressiveness of SPARQL, since only a subset of the language is mapped to form structures. Therefore, another paradigm is supported:

Query Translation An innovative approach to make structured search usable in practice is the automated translation of a keyword query into a set of structured candidate queries. Technically, this is realized by mapping every keyword to elements of the structured data and deriving possible structured queries that lead to "reasonable" results, letting the user select the query that comes closest to his information needs. Matching of keywords happens on both instance and schema level, which means that a query is suggested if all the keywords match on elements of a subgraph, no matter if these elements are RDF entities or links between these entities. We omit the technical details here, referring the interested reader to [13] for the theoretical background of this search approach.

As an example, in the enterprise cloud context the translation of the keywords "LUN size NetApp" could lead to a structured query looking for all LUNs on a NetApp filer, together with their respective size. While query translation resembles the form based search paradigm, it is much more flexible and may help to discover relations between data items the user did not even know existed.

Exploration and Analytics We offer different ways to visualize and explore the results obtained from user-defined queries. Going beyond widgets that display the results in form of simple result tables, we also provide widgets for displaying the results in form of charts, diagrams or timeplots (e.g. for historic data). Queries and their visualizations can be stored and attached to resource pages, making it possible to create expressive dashboards and reports that visualize different dimensions of the underlying resources. It is even possible to create query-based page templates for individual resource types, to be instantiated dynamically when loading the page for an instance of the respective type.

Analysis of Historical Data One key feature of the Intelligence Edition is the support for historical data management, which is made possible by regularly storing snapshots of the system data in a queryable history database. With this feature at hand, admins and CIOs can track developments and changes in the data center over time. Given that the historical data is fully queryable, one could e.g. ask for hosts with constantly low CPU usage, and in response redistribute CPU resources in order to optimize the data center's resource utilization.

Visual Data Exploration The eCloudManager Intelligence Edition can also be accessed using Microsoft Pivot[3] as a frontend. This opens up a whole set of new data interaction paradigms, allowing the effective visual exploration of very large datasets by filtering the results with faceted search, drilling down a heterogeneous intermediate result set by any attribute and grouping a result set by different aspects. A data center provider may start with an overview of all the Virtual Machines in his data center, then decide only to display those VMs with the power state “on”, apply another filter to only see those VMs having between two and five LUNs attached, and finally group these VMs by their state to see the set of VMs that at the moment are in an erroneous state. Such an approach could bring new correlations to light: The data center provider might notice that all the erroneous VMs have attached LUNs on the same physical host. This host should probably be checked, it could well be the source of the problems.

7 Experiences and Lessons Learned

The eCloudManager has been in use productively since 2008 in various enterprises. For instance, an early setup was done at a large software development organization of about 1500 developers to provision a sufficient number of test systems and landscapes for all required development configurations and scenarios. Previously this had been impossible in terms of an overwhelming number of configurations and changes to be administered, with systems regularly needing more than 4 weeks to set up, and “shot” systems often needing a week to be reverted to a functioning status. With the eCloudManager it was possible to use large-scale rapid provisioning of complete application landscapes in minutes.

In another deployment at SAP Value Prototyping, which maintains data centers in both Walldorf (Germany) and Cupertino (USA), in one data center alone it has become possible to support more than 875 virtual machines concurrently, delivering a total of 198 SAP Virtual Landscapes to customers.

Clearly, the most immediate and measurable benefits of using the eCloudManager platform result from the management, provisioning and automation features that allow cutting down on provisioning time and reducing the complexity of the support environment. As these features were not the focus of this paper, we refer to [7] and [1], where we have reported on experiences and results in these dimensions. Instead, we here want to discuss experiences related to the use of semantic technologies within the eCloudManager Intelligence Edition.

The first aspect we would like to discuss are the experiences with our data integration approach. The flexibility of the RDF-based integration proved to be a significant advantage in the heterogeneus environments of the enterprise data centers. Using the standard data providers of the eCloudManager, a typical setup including an integration of the relevant data center resources could be performed by the data center administrator with little manual effort in less than a day. In previous deployments, the initial integration of custom sources, which required

---

[3] `http://www.getpivot.com/`

the instantiation of custom data providers and extensions to the ontology, have been performed by the developers of the eCloudManager. While an integration based on a fixed schema and an a priori limited set of supported APIs would have required significant custom development, with our approach custom sources could be integrated within a matter of a few hours rather than days, primarily because additional data providers could be scripted without changes to the code base. Based on the experiences so far, we expect that in the future also data center administrators themselves will be able to integrate custom sources.

Concerning the interaction with the data center, the users reported several qualitative improvements. While previous management solutions let the resources in the data center appear as unconnected silos, for the first time users are now able to see and explore the data center as an integrated whole, manifested in a connected data graph that can be browsed and explored without boundaries. The ability to perform structured queries was perceived a huge efficiency gain, as previously creating integrated reports across management APIs and databases was considerable effort that involved performing queries against and compiling the results manually. The most enthusiastic reactions were received in response to the visual exploration with the Pivot interface, which provides an unprecedented user experience. With respect to user-generated content, we found that semantic wikis provide an adequate interaction paradigm to generate documentation augmented and linked with structured data. As administrators are used to maintain wikis with documentation anyway, the barrier to add structured annotations is low, while at the same time the benefits are immediately visible.

The final observation concerns scalability and performance of the system. In our current projects – which already include rather large enterprise cloud environments – we have to deal with up to 5000 data center objects per installation, where for each object the number of statements is in the range of tens. When also managing historic and aggregated data, the amount of data increases roughly by a factor of 10. This results in a total of 10^5-10^6 statements – an amount of data that has not posed a serious challenge for the underlying data store nor any other component of the system. As a result, we were able to realize real time ad hoc queries and analytics with sub-second response times.

8 Related Work

In the domain of data center (and more recently enterprise cloud) management solutions, major vendors have commercial offerings, such as IBM Tivoli, HP OpenView or VMware vCenter, yet most commercial systems are far from having a truly integrated data source and appear more like a bundle of individual standalone products. To our knowledge, the fluidOps eCloudManager product suite is the first commercial offering based on semantic technologies to enable an open, vendor-independent integration of heterogeneous data center resources.

The idea of using wikis to improve collaboration and knowledge management in the enterprise is of course not new (see e.g. [2]). Recent works also explore the use of semantic wiki technology in the corporate context. For instance, [8] presents an approach to the collaborative management of systems monitoring information based on Semantic MediaWiki. From a more general, wiki-independent perspective, the benefits of semantic technologies for enterprise data management have been exemplarily discussed in the domain of customer data management in [11]. All these works are complementary to our work, which has its focus on the domain and challenges of enterprise cloud management.

The topic of intelligent analytics for data center operations – or *data center intelligence* – has recently attracted significant interest. As an example, CIRBA[4] offers analytics software that enables organizations to support intelligent planning and management of physical and virtual infrastructure. It helps answer the questions of where to place workloads and how to allocate and configure resources. While the task of data center planning goes beyond the currently available features (cf. future work), the eCloudManager provides integrated access to the relevant technical and business data, including the historical data.

9 Conclusions and Future Work

We have shown how semantic technologies are applied in addressing some of the key challenges in managing enterprise cloud environments. Summarizing the main results, the approach of RDF-based data integration allows us to deal with the highly heterogeneous and changing set of resources encountered in enterprise data centers. Semantic wikis provide an end-user oriented interface for creating structured and unstructured annotations, supporting the main use cases for documentation and knowlegde management, seamlessly integrating automatically obtained data with user-generated content. This data can be searched, explored and analyzed without system boundaries, supported by state-of-the-art techniques of semantics-based information access. Having summarized the results, we now conclude with a discussion of items that remain on the research agenda.

Policies Cloud operators typically define business policies on a high level, which are subsequently monitored by software or taken into account when decisions are being made. In order to be successful, cloud providers need to be able to adapt policies upon changing market requirements or new competition coming into the equation. Changing policies quickly is only possible when the gap between high level business policies and low level implementation is not too big. Future work may consider how to map business policies into rule based systems.

Reasoning Recalling the discussion in Section 3, we currently use the eCloudManager's OWL 2 ontology primarily to define domain, range and cardinality restrictions. As explained earlier, this schema information is then used at runtime, e.g. for guided completion of missing information (cf. Section 5) or schema-agnostic search (cf. Section 6). Going beyond the pure use of schema information, we are planning to investigate in how far the use of OWL (and possibly, advanced features) and OWL-based reasoning can help to monitor system health and detect misconfigurations in the data center.

---

[4] `http://www.cirba.com/`

Complex Event Processing A closely related area where we expect interesting opportunities for future work is complex event processing. Using our provider concept, we periodically receive data from sources distributed across the whole enterprise, and it is a challenging task to define, identify and react to globally meaningful event patterns. The topic seems particularly interesting in the context of reasoning and data semantics, where logic-based approaches like in [4] may be a good starting point for future work.

References

1. Whitepaper: Making the case for the private sap cloud. Technical report, fluid Operations, 2010. `http://www.fluidops.net/download/VLM-Whitepaper.pdf`.
2. Espen Andersen. Using wikis in a corporate context. *Handbuch E-Learning*, 2005.
3. Renzo Angles and Claudio Gutierrez. The expressive power of sparql. In *International Semantic Web Conference*, pages 114–129, 2008.
4. Darko Anicic, Paul Fodor, Nenad Stojanovic, and Roland Stühmer. An approach for data-driven and logic-based complex event processing. In *DEBS*, 2009.
5. Saartje Brockmans, Peter Haase, Pascal Hitzler, and Rudi Studer. A metamodel and uml profile for rule-extended owl dl ontologies. In *ESWC 2006*, 2006.
6. Jeen Broekstra, Arjohn Kampman, and Frank van Harmelen. Sesame: A generic architecture for storing and querying rdf and rdf schema. In *International Semantic Web Conference*, pages 54–68, 2002.
7. Tim Bussiek, Andreas Eberhart, Peter Haase, Sebastian Schmidt, and Ulrich Walther. Landscape-as-a-service in enterprise clouds. In *Submitted to the Third International Conference on Cloud Computing (IEEE Cloud)*, 2010.
8. Frank Kleiner, Andreas Abecker, and Sven F. Brinkmann. Wisymon: managing systems monitoring information in semantic wikis. In *Int. Sym. Wikis*, 2009.
9. Markus Krötzsch, Denny Vrandecic, and Max Völkel. Semantic mediawiki. In *International Semantic Web Conference*, pages 935–942, 2006.
10. Alexander Lenk, Markus Klems, Jens Nimis, Stefan Tai, and Thomas Sandholm. What's inside the cloud? an architectural map of the cloud landscape. In *CLOUD '09: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, pages 23–31, Washington, DC, USA, 2009.
11. Li Ma, Xingzhi Sun, Feng Cao, Chen Wang, Xiaoyuan Wang, Nick Kanellos, Dan Wolfson, and Yue Pan. Semantic enhancement for enterprise data management. In *International Semantic Web Conference*, pages 876–892, 2009.
12. Borja Sotomayor, Rubén S. Montero, Ignacio M. Llorente, and Ian Foster. Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing*, 13(5):14–22, 2009.
13. Thanh Tran, Tobias Mathäß, and Peter Haase. Usability of keyword-driven schema-agnostic search. In *ESWC (2)*, pages 349–364, 2010.
14. Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner. A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.*, 39(1):50–55, 2009.
15. Markus Winter. Data center consolidation: A step towards infrastructure clouds. In *CloudCom '09: Proceedings of the 1st International Conference on Cloud Computing*, pages 190–199, Berlin, Heidelberg, 2009. Springer-Verlag.