About Monitoring in a Service World

Barbara Pernici(✉), Pierluigi Plebani, and Monica Vitali

Politecnico di Milano, piazza Leonardo da Vinci 32, 20133 Milano, Italy
{barbara.pernici,pierluigi.plebani,monica.vitali}@polimi.it

Abstract. Monitoring of services is becoming more and more common to ensure the quality of applications and to provide the required level of service. Nowadays, the technology needed for supporting monitoring and, as a consequence, also monitoring data are widely available. On the other hand, new challenges and research issues arise concerning the design of the monitoring infrastructure, to provide the relevant information both to users and service providers, and their use, and in particular the analysis of monitored data. This paper discusses the main aspects of monitoring, and the use of monitoring data, focusing on event identification and the evaluation of the actions and resources needed to maintain the required level of quality of service. Research challenges related to modeling and using monitoring data are discussed.

Keywords: Quality of service · Monitoring · Event identification · Quality requirements

1 Introduction

Services based on information technology are becoming more and more widespread to support different types of applications. The cloud computing paradigm has introduced different levels of services, introducing the concept of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [15], and additional layers and service-based approaches are being proposed, such as for instance Business Process as a Service (BPaaS) [5] and many more. Intrinsic to the use of a service-based paradigm, there is the need of regulating the interaction between the service provider and the service consumer. Such interaction requires that a service is provided according the functionality and quality of service levels agreed between the two parties. Such an agreement can be either explicitly or implicitly defined, in terms of expectation from the users of a given service, and it can make the difference between similar alternative services in terms of user acceptance and therefore concerning the success of the service itself.

As a consequence, there is an increasing need of data about the actual conditions is which a service is provided, i.e., Quality of Service (QoS). These data

B. Pernici—The present paper presents an extended view from the keynote presentation of the author at SmartGreens 2016.

M. Helfert et al. (Eds.): SMARTGREENS 2016 and VEHITS 2016, CCIS 738, pp. 3–23, 2017.
DOI: 10.1007/978-3-319-63712-9_1

are needed both on the provider and on the consumer side. On the consumer side, data about the functioning conditions of a service are needed to verify if a service is working and if its expected quality properties are the expected ones. This information can be used to select the best available services first, and to verify their functioning conditions during use, and be the basis for formal contractual agreements between service consumers and providers (Service Level Agreements - SLA). On the providers side, in addition to establishing contractual conditions, the data about provided services can be useful to take corrective measures in case the level of quality varies or could be improved.

This aspect is particularly important if the number of consumers for a given service is variable over time, in particular when services are on demand, scalability and elasticity are expected, and therefore the service requires a variable number of resources to satisfy consumers requests.

As discussed in [1], monitoring is needed at different abstraction levels: high-level monitoring provides information about the status of the virtual platform, collected by providers or consumers at the level of middleware, application or users, by the parties themselves or by third parties; low-level monitoring is performed at the providers side to collect specific information at the hardware level, operating system, middleware, network infrastructure and the facility supporting the IT infrastructure. As a consequence, probes needed to collect the information can be located in connection of the different layers of the service infrastructure.

Once the monitoring data are collected and analyzed, the information deriving from the monitoring activity can be used to perform three main types of actions:

- *Adaptation actions*: the system can be adapted, reconfigured, using resources in different quantities and in different ways, in order to provide the requested service and at the requested level.
- *Flexibility support*: the system is capable, possibly also with the support of human intervention facilitated by the monitoring infrastructure, of coping with changing requirements and modes of operation, allowing variable loads and variable levels of service according to different contexts of execution.
- *Awareness support*: the system is capable of analyzing data and of presenting the information to customers and providers in a synthetic way, in the form of dashboards that provide an overview of the status of the system, possibly also associated with alarms and specific warning actions.

The use of monitoring data is becoming more and more widespread in a variety of systems in which adaptivity, flexibility, and awareness support are foreseen.

In the following, we give an overview of some relevant scenarios illustrating them with some examples:

- *Adaptive processes*: since the early days of web service technology, context awareness has been the basis for adapting process-based service compositions: in the PAWS system [3], context is specified on the basis of quality of service parameters, used to evaluate the global quality of the service composition, in Process-Aware Information Systems (PAIS) the execution of flexible processes is designed to be variable in different situations [19].

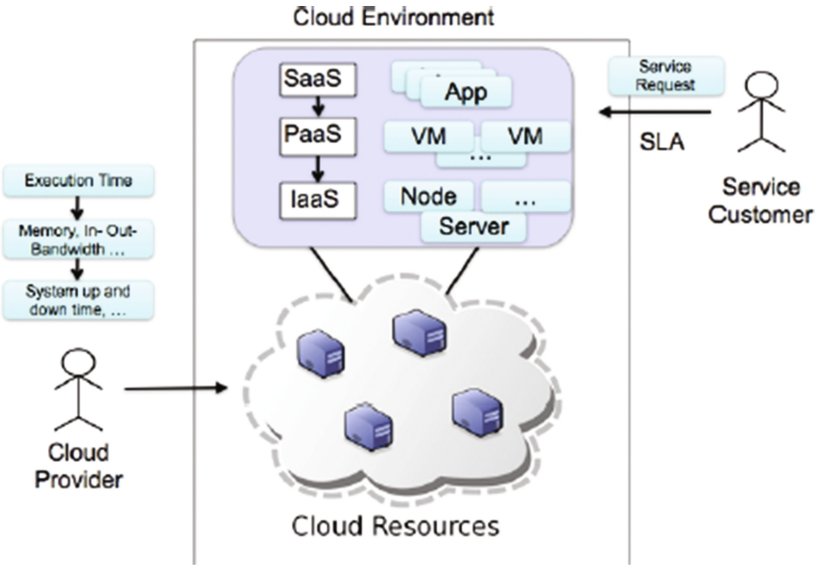

Fig. 1. QoS in services example.

- *User awareness*: several awareness systems, such as, for instance, for utility systems, such as in the case of energy consumption, monitored data are used to inform consumers about their behaviour patterns and to create situations in which consumers are prompted to change their previous behaviour. Tools for facilitation behaviour changes in consumers may be based on serious games and social interaction [4].
- *Cloud service provisioning infrastructures*: as it will be discussed in Sect. 3, monitoring infrastructures are always provided in cloud computing environments, in which QoS is one of the essential elements. In such environments monitoring can present different characteristics in terms of frequency and cost, and it can be used for regulating not only the interactions between providers and consumers, but also the internal use of resources by providers; in Fig. 1, the high-level monitoring and low-level monitoring needs for consumers and providers respectively are shown. Specific uses of monitoring data can be found in methods for improving energy efficiency [10,22] or for reducing the environmental impact of cloud computing [9].
- *Internet of Things*: the availability of monitoring data from sensors is one of the characteristics of the Internet of Things (IoT). Several applications are being developed and others are envisioned for the future. For instance, one application domain can be found in home care, as illustrated in Fig. 2, which shows the architecture for home care developed in the Attiv@bili project [23]. In this case the monitoring data are captured by sensors in smart watches and the infrastructure must guarantee the correct delivery of the monitoring data to all interested applications and interconnected systems.

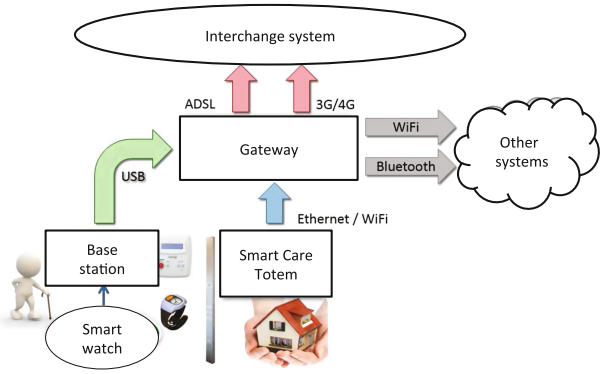

Fig. 2. Smart monitoring devices architecture in Attiv@bili.

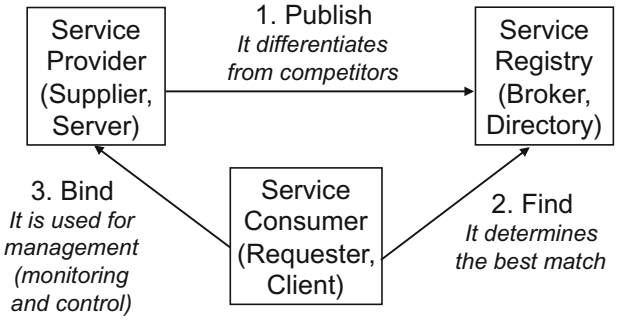

Fig. 3. QoS in a Service Oriented Architecture.

As shown in the previous paragraphs, monitoring can be performed on different types of observed components and for several purposes. In this paper, we discuss some critical aspects of service monitoring and present some research challenges. First of all, Sect. 2 introduces the concept of monitoring as a system that needs to be designed according to the goals to be achieved with the monitoring activity. In Sect. 3, we introduce the main elements at the basis of a monitoring system, including the need of specifying the quality dimensions and measures, and evaluating the quality of the monitoring data. In Sect. 4, we analyze more in detail the different techniques to use the monitoring data and their potential and implications. Finally, in Sect. 5, we present and discuss some research challenges concerning monitoring requirements and analysis and use of monitoring data.

2 Designing Monitoring Systems

As introduced in the previous section, QoS is an important aspect in Service-Oriented Architectures (SOA). Referring to the traditional SOA, quality is

relevant in all phases, as illustrated in Fig. 3. The provider of the service has to declare the quality level which can be provided for the service, stored in a QoS-aware service registry [7]. QoS properties can be evaluated during the service selection phase, to determine which is the best service according to the consumer's goals, and Service Level Agreements (SLA) are defined in the binding phase, specifying the conditions for service provisioning that have to be satisfied during service execution.

Once the provider and the consumer agreed on *what* to monitor, the design of a monitoring system requires to focus on *how* to monitor. Without entering into the details of a monitoring infrastructure, as discussed in [21] a monitoring system has to provide two main modules: the interceptor and the logger. While the former is in charge of transparently capture the information exchanged by the service provider and consumer, the latter is responsible for storing such messages. As also proposed in [8], depending on where the interceptor and the logger are deployed, several configurations are possible (see Fig. 4):

- *Monitor in the middle*: the monitoring system is provided by a third party which is in charge of intercepting and storing information about the execution of the service. An API is provided to allow both consumers and providers to access to the information stored. This configuration is mainly focused on monitoring the information exchanged from which some of the QoS attributes can be inferred. For this reason, information about the status of the provider infrastructure on which the service is running are not available.
- *Monitor at the consumer side*: the monitoring system is under the control of the consumer. In this way, it is possible to collect all the information about the QoS that is considered relevant for the consumer. Similarly to the previous configuration, QoS attributes that can be monitored or inferred must be based on the information exchanged. As the data collected by the consumer can be considered private, no API is usually provided.
- *Monitor at the provider side*: the monitoring system is under the control of the provider. In this case, the provider can collect information about the status of the modules and devices on which the execution of the service is based on. Moreover, monitoring information can be customized for the different customers of the same service. This improves the knowledge that a provider can obtain from the service execution but, at the same time, also increases the effort required to store and manage the ever increasing amount of monitoring information. An API is made available to allow the customers to access to the monitoring information of their interest.

For the sake of simplicity, we do not consider additional configurations where the interceptor and the logger are deployed on different sites. Moreover, where not explicitly defined, hereafter the monitor at the consumer side configuration is assumed to be adopted.

As a consequence, monitoring is to be considered in the design of applications, since it is essential for the correct functioning of service-based applications, to ensure that service provisioning satisfies the conditions defined in the agreement between providers and consumers.

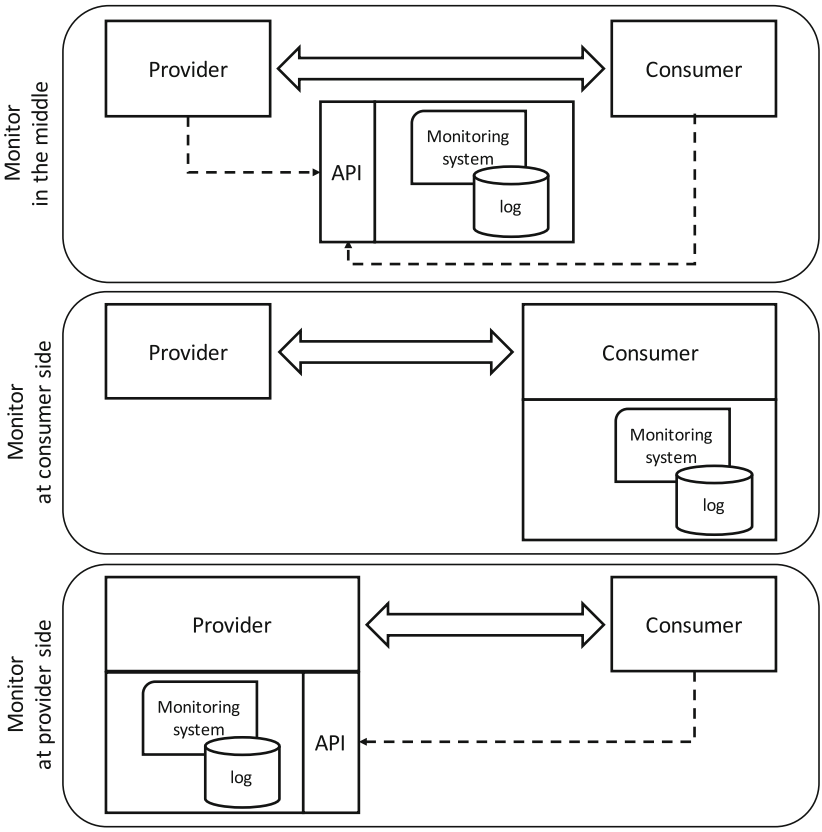

Fig. 4. Monitoring systems configuration deployments.

For instance, in the S-Cube project [2,20], a life cycle for service-based applications has been defined including two main iteration cycles, as illustrated in Fig. 5: the evolution and the adaptation cycles. In the evolution cycle, service-based applications are designed/re-designed according to requirements, both in terms of functionalities to be provided and in terms of possible adaptation actions to be used at run time in the adaptation cycle during the execution of the adaptive application. Monitoring is an essential component during the execution of applications, as it provides the information needed to identify adaptation needs and eventually the requirements for system evolution. In order to provide the needed information to identify adaptation needs and to select the adaptation strategies, also monitoring becomes an element of the design/evolution cycle, since it is necessary to specify the elements to be monitored in order to have the required monitoring information to take appropriate decisions. Such a service life cycle allows the service provider to maintain the agreed conditions for service, continuously adapting service executions depending on the variable context of

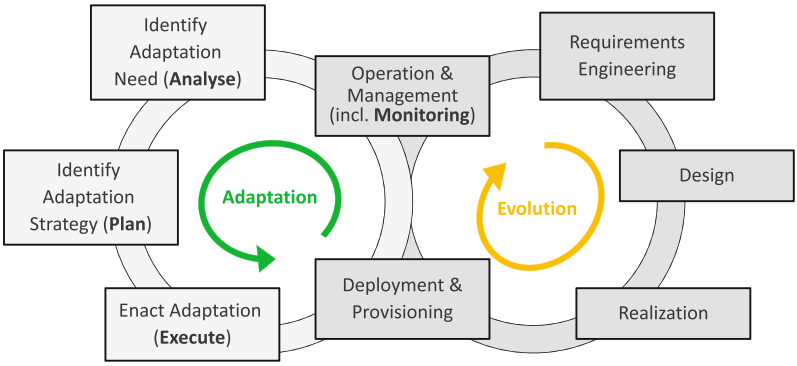

Fig. 5. S-Cube life cycle [2,20].

execution, and evolving the system when adaptation is not sufficient any more to guarantee the required QoS.

Considering monitoring as an element of design implies that the elements to be monitored must be specified and their relevant QoS characteristics defined. In addition, monitoring must support the evaluation of conditions, to be able to identify adaptation needs in a timely and accurate way.

As a consequence, there is a need for a a systematic approach to monitoring, and in particular in the following of the paper we focus on two main issues to be considered during the development of a monitored service:

- *What*: models to specify monitoring elements, their characteristics, assessing the implications of monitoring choices (Sect. 3).
- *Usage*: how monitoring data can be used to identify adaptation needs, critical situations, considering that the monitoring actions are also an overhead during the execution of the system, and therefore a balance must be found between the need to collect and analyze monitoring data and the amount of resources needed for these activities (Sect. 4).

3 Elements of Monitoring

Service monitoring requires to take into account two main aspects: monitoring the *efficiency* and monitoring the *effectiveness* of the service. In the former case, the goal is to focus on the non-functional aspects of a service, especially focusing on how well a service is providing its functions to the consumer. Performance, security issues, and privacy are typical examples of aspects to be considered. In the latter case, the goal is to focus on the functional aspects of a service checking if the exchanged data are correct; this requires the adoption of *data quality* (a.k.a. information quality) techniques to realize if, when invoking the operations exposed by a service, the results are not as expected or the service is not able to recognize non-valid input data. Efficiency and effectiveness of

monitored service usually go under the umbrella of the QoS, with the modeling techniques discussed in Sect. 3.1.

In addition to the QoS, another perspective needs to be taken into account: the *quality of monitoring*. As mentioned in Sect. 2, the monitoring activity is performed by a service, thus being a service, it is important to evaluate its quality, i.e., the quality of the data collected through the monitoring. In fact, the monitoring system produces streams of data from diverse, and often heterogeneous sources. These data are fundamental for figuring out possible misbehavior of the service while it is running. In addition, data could be subject to further analysis to identify patterns and trends that are important to improve the service. Section 3.2 focuses on this aspect discussing which are the attributes that characterize the quality of the monitoring.

3.1 Modeling the QoS

Modeling QoS means providing a tool for the consumer of the monitored service to realize if the service is running as expected. For this reason, we need to specify two aspects: how the service is behaving, i.e., the status of the service, and what the consumer is expecting from the service. In other words, on the one side the consumer specifies the *monitoring requirements*, i.e., the information about the status of the monitored service that he/she is expecting to obtain from the monitoring system. On the other side, the monitoring service is defined in terms of *monitoring capabilities*, i.e., the information that the monitoring service is able to provide. Aligning these two perspectives is often a cumbersome task due to the intrinsic characteristics of the QoS. Indeed, as also stated in the ISO 9216 [13], QoS is a *subjective*, *domain-dependent*, and *multi-dimensional* concept.

Starting from the subjectivity, each consumer of the monitored service could be interested on different aspects about the functioning of the service to evaluate its quality. Some consumer might be more focused on performance, some other on cost. As a consequence, the QoS model needs to be flexible enough to make a custom definition of QoS possible. Moving to domain-dependency, the monitoring requirements for two services living in two different domains, even if they are specified by the same consumer, may result on different definitions of service. For this reason, the QoS model should not include a pre-defined set of dimensions to be used in the QoS definition. On the contrary, the QoS model needs to be extensible, i.e., it should be able to consider additional quality aspects that could be relevant only for a specific domain (e.g., refresh rate for a shared storage service). Finally, QoS is multi-dimensional: it cannot be defined by a single aspect, but it requires the modeling of different perspectives, each of them related to the others. Thus, the QoS model must permit the definition of relationships among the dimensions used to specify the different quality aspects.

To capture all these requirements a QoS model should be compliant to the QoS metamodel proposed in [14] and also reported in Fig. 6. Focusing only on the most relevant elements, the `Service Quality Offer` and the `Service Quality Request` express the monitoring capabilities and requirements, respectively. As they need to be compared to realize if a monitoring system can satisfy the

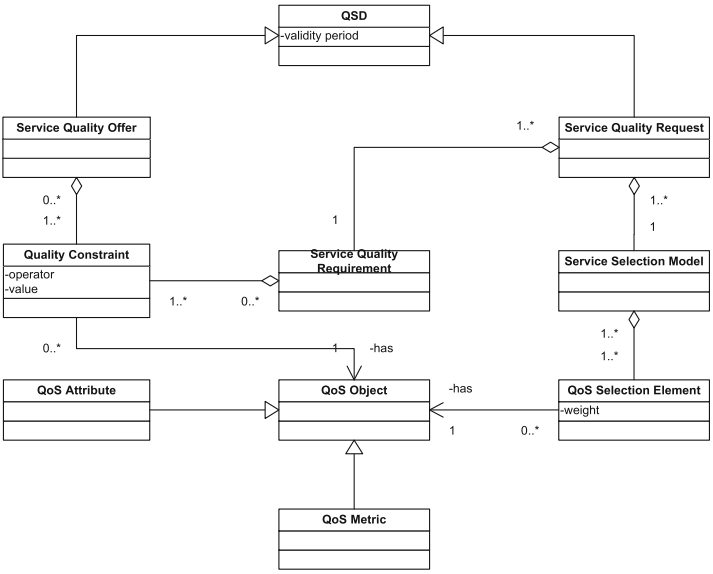

Fig. 6. Service quality meta-model [14].

consumer expectation, they must be based on the same model, i.e., the QoS Definition (`QSD`), although it is used in a different way. The consumer expresses requests listing and ranking the `QoS Objects`, where the ranking is obtained using weights expressing the order of importance among the QoS objects. On the other side, the monitoring service lists the QoS objects that is possible to measure making their values available to the consumer. Based on this structure, the QoS object represents a central point in QoS modeling as the requirements and capabilities base their definition on this element. Being a placeholder for the real QoS aspect to be considered, the resulting QoS model ensures the expected flexibility. Indeed, the `QoS Attribute` and the `QoS Metric` that could specialize a `QoS Objects` represent what is really asked/offered by the consumer/provider. More in detail, as shown in Fig. 7, a `QoS Attribute` defines an aspect that can be relevant for the monitored service (e.g., response time, availability). These attributes can be logically grouped in `QoS Categories` (e.g., security, performance) that give a high-level representation of how the quality of a service can be defined to also increase the readability of the model. Finally, each attribute needs to be measured and one or more `QoS Metrics` can be defined. For instance, the availability of a service can be computed using the classical metric that computes the ratio between the amount of time in which the service is available and the total amount of time.

As an attribute is related to one of the aspects defining the behavior of a service, to have a complete view about a service not only many attributes can be requested by a consumer and many could be offered by the monitoring service provider, but also relationships among the attributes may exist. For this reason,

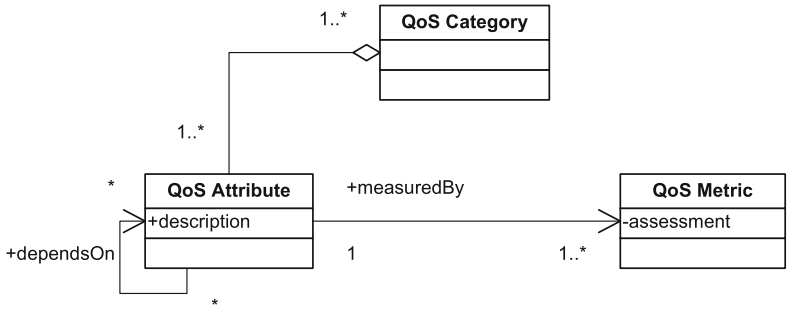

Fig. 7. Monitoring elements [14].

as reported in Fig. 7, dependency relations among attributes must be considered. In some cases, the dependency is clear and exists according to the definition of the attribute itself. For instance, the energy consumed by a service while running depends, among the others, on the amount of CPU and memory used [9]. In some other cases, hidden relationships may exist and, as discussed in Sect. 4.2, some approaches are required to discover them.

3.2 Modeling the Quality of the Monitoring

Data collected by the monitoring system are a valuable source of information for both the provider and the consumer. The former can realize if there are problems while running the service and, in case, space for improvement can be sought. The latter can figure out if the service is working as expected or, comparing the monitoring data, coming from equivalent services, can decide which is the best one.

In any case, it is important that data collected by the monitoring system have a good quality. Indeed, poor data quality may cause false positives and false negatives in detecting anomalous behaviors. Moreover, especially when very little variations for some attributes have significant impact on the evaluation of the service, the accuracy of the data is fundamental.

As any type of quality, also data quality definition – similarly to QoS as discussed before – is multi-dimensional, subjective, and domain dependent. In particular, four main attributes are really important: timeliness, accuracy, completeness, and availability. These attributes are defined in [25] and reported in [6] as:

- *Timeliness* refers to "the delay between a change of the real-world state and the resulting modification of the information system state."
- *Accuracy*: "inaccuracy implies that the information system represents a real-world state different from the one that should have been represented." Inaccuracy refers to a garbled mapping into a wrong state of the system, where it is possible to infer a valid state of the real world though the not correct one.
- *Completeness* is "the ability of an information system to represent every meaningful state of the represented real-world system." Of course, completeness is tied to incomplete representations.

– *Availability* of data is the extent to which data (or some portion of it) is present, obtainable, and ready for use.

Obtaining a good level of quality for the monitoring data according to these attributes depends on many factors. Monitoring QoS attributes requires the installation of probes that are in charge of reading the status of a phenomenon related to the attribute and to convert it into a value. Probes can be implemented as hardware or software. For instance, in a data center the monitoring of the power consumption of a machine requires the installation of PDU (Power Distribution Units) able to detect the amount of power and to send this information to a monitoring system. In this case, possible errors can come from a not proper calibration of the instruments.

Moving to another example, if the attribute to be monitored is the power consumption of a service installed on such a machine, then the data obtained through the PDU is only one of the composing elements [9]. Indeed, as discussed above, the power of a software process (associated to the service) depends on the amount of power consumed by the machine, the fraction of CPU used by the process, the amount of memory and the I/O bandwidth. For this reason, this attribute requires the presence of a software probe that collects all the required information and calculates the final value. Here errors depend on factors like the precision of the probes, the approximations done during the computation, and the reliability of the formulas adopted. Given this complexity, monitoring data are thus prone to systematic errors which affect the quality of the data returned by the monitoring system and the minimization of these errors requires higher costs. High quality probes are more expensive, as well as more precise computations could increase the resource demanding for the monitoring system and thus the cost.

A proper balance between the cost of designing and running the monitoring system and the quality of the monitoring data is crucial. In addition to the aspects introduced above, particular emphasis can be reserved to the cost of storing the monitoring data. Especially when the monitoring system is installed at the provider side, the amount of monitoring data depends on the number of services, the number of attributes for each service, and how frequently these attributes are measured. As an example, Cloudera[1] offers more than one hundred categories of metrics, with a one minute sampling rate, with the possibility to define aggregation functions. Indeed, more attributes and more frequent sampling time result in higher quality of monitoring data and higher costs for storing all these data. For this reason, especially for cloud providers, different monitoring services are proposed with different costs. For instance, Amazon CloudWatch[2] offers both a basic monitoring service where pre-selected metrics are made available at five-minute frequency with no additional cost, and a detailed monitoring where the set of metrics is the same but at one-minute frequency and with an additional cost. Also

[1] http://www.cloudera.com/documentation/enterprise/5-6-x/topics/cm_metrics.html.

[2] https://aws.amazon.com/it/cloudwatch/.

Paraleap CloudMonix (formerly known as AzureWatch)[3] offers both the possibility to monitor an unlimited set of metrics but at ten-minutes frequency with no additional cost, and at one-minute frequency with a fee.

Based on this scenarios, taking into account the quality of monitoring data introduces also a new dimension for the service provider selection. Indeed, when the same service is offered by several service providers, the selection is usually based on the QoS. Introducing also a proper evaluation of the quality of monitoring data allows the service consumer to select the service provider that also provides the most reliable and accurate monitoring data that can be useful for ex-post analysis. About this scenario, [12] proposes an approach that optimizes the quality of monitoring considering the accuracy of the quality attributes, the coverage and the extensibility of the monitoring system, while respecting budget constraints of the consumer.

4 Use of Monitoring Data

As mentioned in Sect. 3, the data collected by the monitoring system can be relevant to ensure the quality of the service provided, to identify issues about the service behavior, and to react to undesired situations in order to restore an effective behavior for the monitored service. In a SOA, an agreement is negotiated between the service provider and the consumer, the SLA in which metrics are identified together with their acceptable and undesired values, referred as Service Level Objectives (SLO) or quality objectives as shown in Fig. 6. The monitoring system is the source from which SLOs can be computed and the satisfaction of the SLA can be evaluated. Two phases of the SLA life-cycle are involved in this activity: the SLA assessment and the SLA settlement [14]. In the assessment SLOs are evaluated and compared with reference values and constraints. The assessment has to be executed regularly and has a period of validity. When a SLO is violated, some repair actions can be taken. This operation is part of the settlement phase in which penalties and rewards are assigned according to satisfactions and violations of the SLA, and actions affecting the SLA evaluation outcome are prescribed.

The main issues related to the exploitation of the monitoring data for assessment and settlements are:

- *Events identification*: not all the collected information is relevant, techniques are needed for isolating relevant events that should be considered for enacting repair strategies (Sect. 4.1).
- *Condition evaluation*: identified events have to be compared with reference values in order to determine if an undesired behavior is occurring, and strategies for bringing the service back to a normal behavior need to be selected (Sect. 4.2).
- *Resources and cost*: monitoring has a cost in terms of computational resources and storage space, but also an economic cost changing with the number of collected metrics, their quality and precision (Sect. 4.3).

[3] http://cloudmonix.com.

In the following of this section, these three issues will be discussed in more detail.

4.1 Identifying Events

The analysis of the satisfaction of SLOs is usually performed through the evaluation of indicators, usually related to the quality perspective (Key Performance Indicators - KPI), but also to other perspectives (e.g., energy efficiency through Green Performance Indicators - GPI). Indicators are assessed through the computation of one or more metrics, associated to them, starting from the data collected by the monitoring system. A set of thresholds can be associated to each indicator, defining which are the desired and undesired values [11]. When an indicator is outside the desired range of values, something should be done to correct this misbehavior. The approach towards the assessment of indicators can be either proactive or reactive. A *reactive approach* considers as an issue only violation of the desired values range. In the *proactive approach*, violations should be avoided by preventing them observing the trends of the indicators values. In order to do that, an alarm set of values is considered between the violation and the satisfaction zone. An indicator whose value is in the alarm zone is not violated, but it is likely to be violated soon. Figure 8 shows the intervals of satisfaction, violation, and alarm of a generic indicator. Each of these regions is defined by two thresholds. Defining which are the best values for these thresholds is not an easy task. In order to define the set of undesired values, it is possible to use as a reference best practices (e.g., the Green Grid Data Center Maturity Model[4]) and consumer requests through the Service Level Agreement (SLA). Also, changing the size of the alarm zone, the approach to violations can change from a more reactive to a more proactive approach, thus making this decision flexible. The threshold definition issue has been discussed in more detail in previous work (e.g., [18]).

Fig. 8. Identifying satisfaction, alarm, and warning zones for indicators.

Classifying the indicators value in one of the zones described before is not enough, since it is also important to understand the severity of the violation. This can be evaluated performing a normalization of the indicators values [17] where each value is transformed into a value in the interval $[0, 1]$. Thanks to this normalization it is possible to compute a complex metric obtained from

[4] http://www.slideshare.net/martinciupa/data-center-maturity-model-white-paper finalv2.

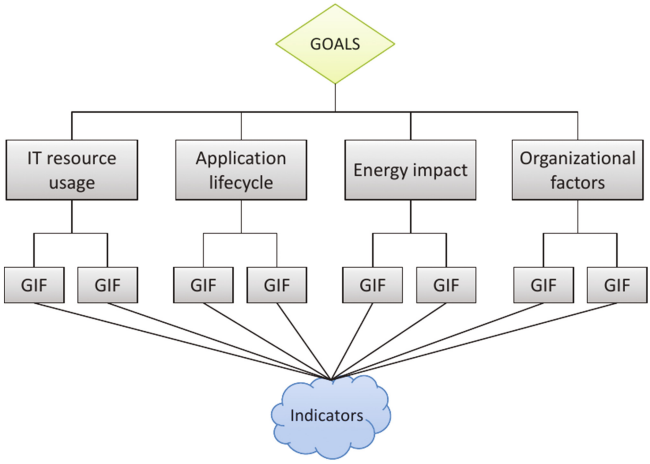

Fig. 9. Indicators aggregation organized in a hierarchical way [17].

the aggregation of several indicators concurring to the satisfaction of a common goal, considering what in Sect. 3.1 was referred to as a QoS category. The aggregation can be performed as a weighted sum function of the selected metrics values, referred as Green Index Function (GIF). Indicators can be hierarchically organized as shown in Fig. 9, in which categories (e.g., IT resource usage, Application lifecycle, Energy impact, and Organizational factors) aggregate indicators through a GIF, and are used to define complex goals.

As shown in Fig. 9, indicators or their aggregation can be considered as goals and can be represented in a goal-oriented model in which goals are indicators satisfaction. An event should be raised when a violation is observed, implementing the adaptation cycle presented in Fig. 5. The adaptation process is composed of two phases: the event creation and the adaptation strategy selection. An event is raised when a significant violation of an indicator threshold has been observed. Not every violation generates an event. In fact, temporary violations can be ignored since they can automatically recover or they can be due to noise. The approach proposed in [16] addresses the identification of significant violations in order to generate events. Through the Integrated Energy-aware Framework (IEF), violations are identified considering their duration over time and their severity, thus generating an event only if the indicator is in the alarm or warning zone for a time exceeding the accepted duration for a violation. An event is thus described by the timestamp in which it has been generated, the value associated to the violated indicator, a direction stating if this value is increasing or decreasing (trend), a significance defining which event occurrences have priority, and a severity computed considering the quantitative importance of the violation.

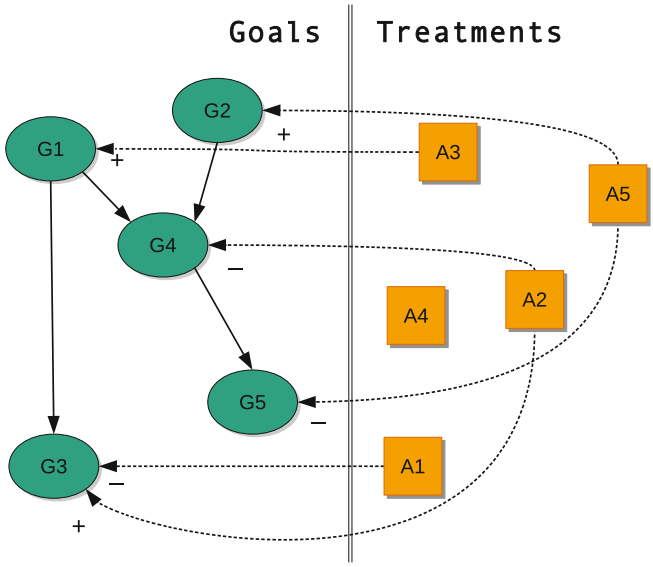

Fig. 10. Modeling adaptation through a goal-based model.

4.2 Condition Evaluation and Selection of Actions

Events are symptoms of misbehavior in the monitored environment and can be used as drivers for adaptation. Evaluation of the conditions over indicators for the generation of events is a complex task. In fact, the evaluation of violations is sensitive to several factors. As discussed before, aggregated indicators can be obtained from the weighted sum of a set of other indicators. In this case we deal with a multi-attribute metric, in which weights can change over time, affecting the evaluation of the aggregated metric. Another factor to be considered is the consumer: different consumers can have different preferences, different stakeholders of the same service can have a different perception of what should be considered as a violation of the constraints. Finally, evaluation can be context-dependent, and some constraints could get relaxed under some circumstances.

Adaptation can be performed by enacting an adaptation strategy, consisting in one or more treatments (or actions), affecting the state of the system. In a goal-oriented approach, it is important to model:

- Which goals are affected by the enactment of an action.
- Which kind of outcome (positive or negative) the action has over the affected goals.

A representation of this action-to-goal model is represented in Fig. 10, where two layers are depicted: a goal layer and a treatment layer.

The goal layer contains the identified goals, modeled as the satisfaction of constraints over indicators. As discussed in Sect. 3 and shown in Fig. 7, direct

and indirect dependencies among metrics may exist. In the goal layer of the model, these relations between indicators are also modeled as shown in the left part of Fig. 10. These relations are important for predicting the outcome of a modification and for conducting what-if analysis. Modeling relations between metrics can be a trivial activity if performed manually by expert. As discussed in Sect. 3.1, dependencies can be directly extracted from the definition of the metric, or can be hidden. For discovering hidden relations, automatic approaches can be employed. In [24], a representation of relations between goals of a goal-oriented model is provided using a Bayesian Network expressing causal dependencies among goal states. The network is automatically computed from the analysis of the information collected through the monitoring system using a three steps approach:

- Learning the structure of the Bayesian Network (which goals have a relation).
- Learning the direction of edges in the Bayesian Network (cause vs effect).
- Learning the parameters of the Bayesian Network (conditional probability tables for each node).

The treatment layer contains all the actions that can be used to affect the system state and it models their effects over the goals. The model can be used to decide which is the best strategy to enact when a violation occurs. Since an action can have both a positive and a negative effect over the indicators, it is important to take into account both direct and indirect effects when taking a decision on which action to enact. This decision is strictly dependent on the context, which is the state of all the goals in the considered system. Since services are executed in a complex and dynamic environment, discovering and modeling action-to-goal relations is a complex task. In [16], the selection of an action is always followed by a history-based analysis addressed to recognize patterns and used to validate the current model and to eventually modify or create new relationships. In [24], the action-to-goal relations are acquired through a continuous refinement algorithm (the Adaptation Action Selection - AAS), which starting from scratch observes the outcomes of action enactments and updates the model considering both the current observation and past observations. The action effect over indicators is associated with an impact value, assessing the probability that applying the action the value of the indicator will change. The selected adaptation strategy should maximize positive effects over violated indicators, while minimizing side effects over the satisfied ones. This algorithm enables self-adaptation when an action modifies its outcome over indicators due to some external noise and is not sensitive to the noise caused by other events affecting the environment of execution during the observation.

When evaluating the effectiveness of an adaptation approach several criteria should be considered:

- *Stability*: the decision taken by the algorithm should bring to a stable system for a sufficient amount of time if no external factors occur. Stability avoids that the algorithm keeps prescribing modifications, that in some cases could

contradict themselves, in search for an optimal solution that may not exist. An example can be migration: if the performance of a service is not satisfying, it is possible to migrate the service in a host with a better nominative QoS. If after migration a significant improvement is not observed, it is not desirable that the algorithm enacts another migration, since migration is costly and introduces a temporary performance degradation itself.
- *Flexibility*: the solution should be responsive to modifications to the service, to the set of selected indicators, and to the constraints above these indicators without requiring a heavy reconfiguration of the adaptation algorithm.
- *Scalability*: the solution should scale linearly with the increasing number of goals and constraints and with the introduction of new adaptation strategies, keeping the computational time for evaluating the system limited and providing a solution in a reasonable time.

These criteria can be used to compare several approaches and to select the one that fits better in the considered context.

4.3 Resources and Cost

Monitoring is an important step for guaranteeing the quality of the provided service and for assessing the SLA satisfaction. However, collecting monitoring data comes at a cost. The trade-off between the advantage of monitoring and its cost should be considered.

In a cloud oriented environment, the monitoring system is managed by the cloud provider who hosts the service. As discussed in Sect. 3.2, different providers can offer a different quality of service according to their resources, but also a different set of monitored information, at a different quality with a different cost. Also, customization of the monitoring service is important for getting full advantage of the monitoring system. Choosing which is the best cloud provider can be difficult. Several aspects have to be considered:

- *Quality of the service*: different cloud providers can offer a different quality for hosting the service in terms of KPIs such as response time and availability.
- *Monitoring offer*: the amount of available metrics can change, also their granularity can be different. Some providers offer only metrics at the physical level, others collect also Virtual Machine level metrics and application level metrics.
- *Monitoring quality*: quality of the monitoring system depends on the precision of the collected information which depends mainly on the sampling time and on the ability to store old data to perform analysis [12].
- *Monitoring extensibility*: some cloud providers offer the possibility of extending the monitoring system with custom metrics both at the hardware level (through the installation of new probes) and at the software level (through the execution of scripts).

According to their characteristics, the monitoring services offered by the different cloud providers have different costs. It is important that this cost does not

overcome the advantage obtained from the monitoring data. Also, the analysis of the monitoring data requires a computational effort which can be directly proportional to the amount of these data. Computational resources have to be employed for this analysis, thus increasing the total cost of hosting the service.

In order to avoid useless costs, the proper set of metrics, together with the correct sampling time, should be selected. How to select which are the metrics that actually depict the state of a service and provide value to the analysis is still an open issue.

5 Challenges and Future Research Work

In this section, we comment about the open issues and discuss challenges and future research work. As mentioned at the end of Sect. 2, the paper has examined two main issues in a systematic approach to monitoring: the "What", i.e., the specification of monitoring elements, and the "Usage", i.e., how monitoring data can be used.

The main question about the *What* issue concerns which monitoring data should be collected. The interesting data depend on the service being provided, for each situation not only the elements to be monitored should be specified, but also which are the available *sources* of monitoring information, possibly more than one; another issue concerns the definition of the *granularity* of monitoring data, and the possibility of viewing the data and analyzing it at different granularities; a third issue concerns the *quality of the monitored data*, which should be assessed to provide a complete information about the available information.

Concerning sources, it should be possible to dynamically create or use new monitoring sources, in particular in services being provided in very variable situations, such as, for instance, in emergency situations. The impact of context for evaluating which monitoring data are necessary, the needed granularity, and the need for further analysis are the elements which need further investigation. In fact, monitoring should not be homogeneous, considering always the same elements, but a focus on situations which need attention should be supported. It is also interesting to provide methods to combine different monitoring sources, to better assess and improve their quality, in particular consistency and completeness. The selection of sources should not consider only data provided by the system being monitored, but also consider information originating from other potentially useful sources. For instance, a process being executed is regularly monitored, but some problems occurring during the process execution may be originated by external causes [23], e.g., bad weather conditions in a delivery service or changed health conditions in a regular home care service. The quest for information from unrelated available sources for additional monitoring data should therefore be further investigated. As a conclusion, we can state that while monitoring systems are a rather mature field, there is still need for research in the field of selection of monitoring variables, both in defining their requirements in a precise way, and with the possibility of coping which changing situations, and also providing support for the identification of external monitoring systems

which can provide additional information to the system being developed for providing service. The problem of finding the right sources of data becomes a relevant issue, including the retrieval of sources, associating meta-data to them to be able to correctly interpret the monitoring data, and considering a value-driven source selection. Once monitoring variables are identified in the different sources, it is also important to understand the relationships between variables, as this information can be useful both for filtering data and for relating information from different sources to get additional insight.

When addressing the *Usage* issue, several aspects still require further investigation. First of all, there is a need to coordinate monitoring activities. In fact, monitoring can have phases, and monitoring activities might have to be dynamically selected. There is an intrinsic variability in systems being monitored, which requires a corresponding variability in the monitoring system behavior. In some situations, monitoring is not always necessary or only a light observation is needed, and there is a strong variability of needs between starting up phases (or when adaptation actions are performed [16]), during regular execution, and during exceptional or crisis situations. In general, there is the need to design the monitoring process(es), not only the monitoring system. The monitoring requirements must be specified, defining the monitoring requirements for different goals, the requirements related to different types of events and risks, defining monitoring actions such as the selection of variables, sampling rates, granularity, sources, and so on, in the different situations. It is also necessary to include in the requirements the specification of the quality parameters for monitored variables, such as, accuracy, completeness, and timeliness.

During the adaptation loop, it is necessary to apply the control policies defined at design time. It must be possible to verify if requirements are satisfied in the given context and to be able to identify the best adaptation strategies if adaptation is needed. Therefore, the design of assessment conditions and of adaptation rules are closely associated to the design of the monitoring system. As a concluding remark, we emphasize the need of coordinating monitoring activities, to investigate monitoring as a process, and to study design criteria for monitoring; in addition, for the specification of monitoring requirements research is needed on informal, semi-formal, and formal models and methods.

Acknowledgements. This work was partially developed within the European COST Action IC 1304 Autonomous Control for a Reliable Internet of Services (ACROSS) and partially supported by the Lombardy Region within the Sistema innovativo di Big Data Analytics project.

References

1. Aceto, G., Botta, A., de Donato, W., Pescapè, A.: Cloud monitoring: a survey. Comput. Netw. **57**(9), 2093–2115 (2013). doi:10.1016/j.comnet.2013.04.001
2. Andrikopoulos, V., Bucchiarone, A., Nitto, E., Kazhamiakin, R., Lane, S., Mazza, V., Richardson, I.: Service engineering. In: Papazoglou, M.P., Pohl, K., Parkin, M., Metzger, A. (eds.) Service Research Challenges and Solutions. LNCS, vol. 6500, pp. 271–337. Springer, Heidelberg (2010). doi:10.1007/978-3-642-17599-2_8

3. Ardagna, D., Comuzzi, M., Mussi, E., Pernici, B., Plebani, P.: PAWS: a framework for executing adaptive web-service processes. IEEE Softw. **24**(6), 39–46 (2007). http://dx.doi.org/10.1109/MS.2007.174
4. Arnone, D., Rossi, A., Melodia, E., Mammina, M., Jenkins, S.: Improving energy awareness integrating persuasive game, feedback, and social interaction into the novel ener-scape application. Int. J. Adv. Intell. Syst. **8**(3&4), 437–447 (2015)
5. Barton, T., Seel, C.: Business process as a service - status and architecture. In: Enterprise Modelling and Information Systems Architectures - EMISA 2014, Luxembourg, 25–26 September 2014, pp. 145–158 (2014). http://subs.emis.de/LNI/Proceedings/Proceedings234/article12.html
6. Batini, C., Scannapieco, M.: Data and Information Quality. Springer International Publishing, Switzerland (2016)
7. Bianchini, D., De Antonellis, V., Pernici, B., Plebani, P.: Ontology-based methodology for e-service discovery. Inf. Syst. **31**(4–5), 361–380 (2006). doi:10.1016/j.is.2005.02.010
8. Cabrera, O., Franch, X.: A quality model for analysing web service monitoring tools. In: 2012 Sixth International Conference on Research Challenges in Information Science (RCIS), pp. 1–12, May 2012
9. Cappiello, C., Datre, S., Fugini, M., Melia, P., Pernici, B., Plebani, P., Gienger, M., Tenschert, A.: Monitoring and assessing energy consumption and CO_2 emissions in cloud-based systems. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp. 127–132, October 2013
10. Cappiello, C., Ho, T.T.N., Pernici, B., Plebani, P., Vitali, M.: CO_2-aware adaptation strategies for cloud applications. IEEE Trans. Cloud Comput. **4**(2), 152–165 (2016). doi:10.1109/TCC.2015.2464796
11. Cappiello, C., Plebani, P., Vitali, M.: Energy-aware process design optimization. In: 2013 International Conference on Cloud and Green Computing, Karlsruhe, Germany, 30 September–2 October 2013, pp. 451–458. IEEE Computer Society (2013). doi:10.1109/CGC.2013.77
12. Fadda, E., Plebani, P., Vitali, M.: Optimizing monitorability of multi-cloud applications. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) CAiSE 2016. LNCS, vol. 9694, pp. 411–426. Springer, Cham (2016). doi:10.1007/978-3-319-39696-5_25
13. ISO/IEC: ISO/IEC 9126–1 Software Engineering. Product Quality - Part 1: Quality model (2001)
14. Kritikos, K., Pernici, B., Plebani, P., Cappiello, C., Comuzzi, M., Benrernou, S., Brandic, I., Kertész, A., Parkin, M., Carro, M.: A survey on service quality description. ACM Comput. Surv. **46**(1), 1:1–1:58 (2013). http://doi.acm.org/10.1145/2522968.2522969
15. Mell, P., Grance, T.: The NIST Definition of Cloud Computing. Technical report, NIST, National Institute of Standards and Technology, U.S. Department of Commerce (2011)
16. Mello Ferreira, A., Pernici, B.: Managing the complex data center environment: an integrated energy-aware framework. Computing **98**(7), 709–749 (2016). doi:10.1007/s00607-014-0405-x
17. Ferreira, A.M., Pernici, B., Plebani, P.: Green performance indicators aggregation through composed weighting system. In: Auweter, A., Kranzlmüller, D., Tahamtan, A., Tjoa, A.M. (eds.) ICT-GLOW 2012. LNCS, vol. 7453, pp. 79–93. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32606-6_7
18. Pernici, B., et al.: Setting energy efficiency goals in data centers: the GAMES approach. In: Huusko, J., Meer, H., Klingert, S., Somov, A. (eds.) E2DC 2012. LNCS, vol. 7396, pp. 1–12. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33645-4_1

19. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies. Springer, Heidelberg (2012)
20. S-Cube Team: S-Cube Network of Excellence final report. Technical report (2012). http://www.s-cube-network.eu/final-report.pdf
21. Seely, S., Lauzon, D. (eds.): WS-I Monitor Tool Functional Specification v 1.1 (2005). http://www.ws-i.org/Testing/Specs/MonitorFunctionalSpecification_Final_1.1.pdf
22. Vitali, M., Pernici, B.: A survey on energy efficiency in information systems. Int. J. Cooperative Inf. Syst. **23**(3) (2014). doi:10.1142/S0218843014500014
23. Vitali, M., Pernici, B.: PiE - Processes in Events: interconnections in ambient assisted living. In: Ciuciu, I., Panetto, H., Debruyne, C., Aubry, A., Bollen, P., Valencia-García, R., Mishra, A., Fensel, A., Ferri, F. (eds.) OTM 2015. LNCS, vol. 9416, pp. 157–166. Springer, Cham (2015). doi:10.1007/978-3-319-26138-6_19
24. Vitali, M., Pernici, B., O'Reilly, U.: Learning a goal-oriented model for energy efficient adaptive applications in data centers. Inf. Sci. **319**, 152–170 (2015). doi:10.1016/j.ins.2015.01.023
25. Wand, Y., Wang, R.Y.: Anchoring data quality dimensions in ontological foundations. Commun. ACM **39**(11), 86–95 (1996). http://doi.acm.org/10.1145/240455.240479