Risk-Based Usage Control for Service Oriented Architecture

Leanid Krautsevich and Aliaksandr Lazouski
Department of Computer Science
University of Pisa
Largo B. Pontecorvo 3, Pisa, Italy
krautsev@di.unipi.it
lazouski@di.unipi.it

Fabio Martinelli and Artsiom Yautsiukhin
Istituto di Informatica e Telematica
Consiglio Nazionale delle Ricerche
G. Moruzzi 1, Pisa, Italy
fabio.martinelli@iit.cnr.it
artsiom.yautsiukhin@iit.cnr.it

***Abstract*—In Service Oriented Architecture (SOA) data belonging to a client (data provider) is often processed by a provider (data consumer). During this processing the data can be compromised. A client wants to be sure that its data is used in the least risky way while is under provider's control. The risk level should be low when access to the data is granted and should remain low during the whole interaction and, maybe, some time after. Therefore, a client has to consider closely various providers and decide which one provides the service with the smallest risk. More importantly, the risk has to be constantly recomputed after granting the access to the data, i.e., usage of data must be controlled.**

In this work we propose a method to empower usage control with a risk-based decision making process for more efficient and flexible control of access to data. Employing this idea we show how to select a service provider using risk, re-evaluate the risk level when some changes have happened and how to improve an infrastructure in order to reduce the risk level.

I. Introduction

Various applications of SOA (Web Services, Grid, and Clouds) are based on the idea of outsourcing some parts of business to third companies. In other words, some data belonging to one entity (data provider) becomes under control of another entity (data consumer). Traditional access control in this case is not a complete solution. The data provider apart from allowing access to its data also has to control if the data is used as it has been negotiated. Moreover, it is hard to make a correct access control decision in the environment where the entities often do not know each other before the interaction. Therefore, controlling *usage* of data (after sending the data) is very important in these settings.

The Usage Control (UCON) paradigm has been introduced recently [1] and since then it is gaining more and more attention of the scientific community [2]. This attention is connected with increasing demand of the modern technologies to control data usage also *after* releasing the data to external parties. Next to its traditional application area of Digital Right Management [3] UCON proved to be very useful in Service Oriented Architectures [4], [5], [6].

Giving sensitive data to another entity is connected with a number of risks that the data will be damaged or become available to untrusted parties. Naturally, the data provider wants to minimise these risks. Every data provider has a list of security requirements and its preferences which it wants to be addressed. The problem is to select the data consumer which guarantees the best protection of the data. Therefore, it is important to define a way for computation of risk connected with outsourcing of the data to a concrete data consumer. Such calculation has to consider that each data provider has its own preferences and not all of these preferences can be addressed well enough by the data consumer.

Furthermore, the data provider has to be sure that the data consumer fulfils its promises, i.e., control the usage of data. Interactions between these parties may last for a very long time and various changes occur during this time. Some of these changes affect the negotiated agreement about access to and usage of the data. The data owner should not revoke the access rights immediately because this is not the best solution from a business perspective. Instead, the data provider should consider the general picture and decide if it can accept the new level of risk.

The risk level also can be used by the data consumer which wants to provide the best service to its customers. The risk level is a good indicator to show that the provided quality of protection is good enough or must be improved. The data consumer needs a simple and reliable way to predict the effect of changes in its infrastructure on a risk level.

A. Main Contributions

- The main contribution of this paper is a method for computing the risk of giving data to a specific data consumer. The main focus is made on attributes for computing risk levels of guaranteed service. Having focus on attributes makes the method very flexible.
- The proposed method is used to help data providers to select the data consumers which guarantee the best protection of data, i.e., which have the lowest risk levels according to lists of specified UCON policies.
- The proposed risk-based and attribute-driven method offers more efficient and more flexible comparison of desirable and guaranteed policies.
- Our approach helps to make a decision about the riskiness of the selected service during usage of the data supporting the idea of continuity of policy enforcement

and mutability of attributes.

- Finally, employing the proposed risk-based and attribute-driven method we present a simple way to select the best improvements for data consumer's infrastructure. This selection is simple, efficient and does not require the values which are hard to find (e.g., effectiveness of security controls [7]).

The paper is organised as follows. Section II shortly summarises important aspects of UCON in SOA. In Section III we present our idea about computing the risk level of a guaranteed service. Then we use the proposed method for selecting the optimal data consumer (Section IV). How to re-evaluate a risk level during usage of data is shown in Section V. We propose a way for a data consumer in order to select the best set of improvements to decrease its risk level in Section VI. Section VII is devoted to related work. We finalise the paper with a short discussion (Section VIII) and concluding remarks (Section IX).

II. Usage Control in Service Oriented Architecture

SOA and UCON have different notations and we start with defining the main actors participating in the interactions that we consider. The main actors in SOA are a *client* which requests for some service and a *service provider* which provides the service. From UCON's perspective [2] a client provides data and a service provider consumes the data. Therefore a client is called *data provider* and a service provider is called *data consumer*. This situation is clarified in Figure 1.

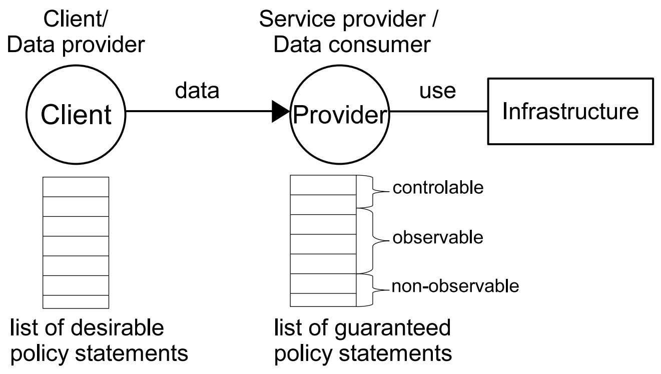

Fig. 1. General SOA/UCON Model.

Each actor has a set of attributes which can be used to define access rules. Also environment itself has attributes (e.g., time, location, etc.) which may affect the decision to grant access to data or not [8], [1].

The whole UCON model can be described using the idea presented in [8] and updated for our case (Figure 2). First, a client starts selecting service providers and selects just one of them (or does not select any) comparing risk levels (see Section IV). Then, the contract between the partners is signed where all usage rules are stated. Now the data consumer can receive the data (once or access the data periodically). During the usage of data when an attribute is updated the risk level is re-computed and re-evaluated (see Section IV). If the risk level is decided to be too high the contract is terminated and access to the data is forbidden (also already received data must be deleted).

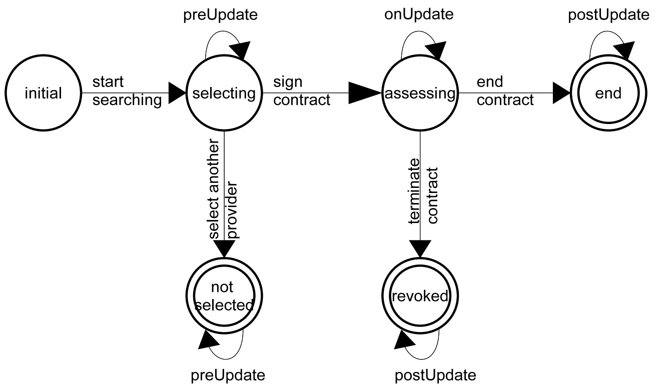

Fig. 2. UCON Model States for SOA

The rules which define how data can (or cannot) be used are called policy statements [8] and can be separated into the three classes: *authorisations*, *obligations* and *conditions*. Authorisations characterise a data consumer and a data provider and are used to decide if access to a resource can be granted. Obligations are actions which must be committed during the usage.

Obligations describe how a data consumer must use the data during the access. Conditions restrict access to the data according to environmental attributes. In the sequel, we do not stress the different nature of the policy statements and attributes, but we only take into account that attributes are used as a core to form policy statements [1].

Example 1: For the policy statement "data must be deleted after 30 days" an environmental attribute "time after request about deletion" is required.

The two aspects which are specific for UCON are [8]: (i) continuity of access decision evaluations and (ii) mutability of attributes. In other words, attributes may change during usage of data and the corresponding decision about further access must be re-evaluated.

There are three ways for a data consumer to assure a data provider that policy statements will be fulfilled [2]: (i) *enforced* statements are enforced by security mechanisms (ii) *observable* statements can be audited by the data provider (e.g., analysing logs) (iii) *non-observable* statements: a data provider must simply trust the data consumer that the policy statements will be followed. As it is shown in [4] current state of practice does not have enough techniques for controlling policy statements. Anyway, in our approach we consider a general case where all ways of assuring data providers are possible.

III. Calculation of risk

Naturally, every data provider wants to minimise the risk related with possible abuse of its data. A data provider has its preferences how the data must be used to minimise the risk of data to be revealed to the third parties, maliciously modified, become unavailable. For this purpose the data provider specifies a list of policy statements. Each policy statement relates to a specific data object and states how the object must be used.

Statements have different importance for a data provider. Violation of some policy statements cannot harm the data provider significantly. Following other policies can be crucial. Moreover, the data provider should consider which policies can be more likely violated by the data consumer while determining the importance of the policies. Only the data provider has this knowledge which is based on the nature of the data and policies.

We assume that the data provider uses qualitative scale to rate the importance of policy statements assigning *high, medium or low* ranks of criticality to each policy statement[1].

Definition 1: If P_d is a set of all desirable policies then we can define a function $rank$ as follows:

$$rank : P_d \mapsto \{low, medium, high\}$$

Example 2: Suppose that a data provider wants two policy statements to be enforced: "the data must be deleted after 30 days" and "the external audit must be performed twice a year". The first policy is not very important for the data provider and has the level of importance "low". On the other hand, it is important for the data provider that the data consumer correctly processes the data, but does not abuse it, e.g., for committing a fraud. Thus, the second policy statement has "high" rank of criticality.

A data provider starts looking for a data consumer which provides the required service and also satisfies the specified policy statements. Obviously, not all statements can be addressed by every data consumer in general case. The client wants to give the data to (authorise) the data consumer which guarantees the best protection, i.e., the data consumer which satisfies the desirable policy statements in the best way.

Policies may be expressed differently while have the same meaning (e.g., expressed with different policy languages). Thus, first of all, the data consumer must find the correspondence between each desirable policy and a guaranteed policy. We do not consider the problem of semantical mapping of policy statements here and refer the reader to the solutions already proposed in the literature, e.g., [11].

The next step is to consider if the desirable policies are addressed well enough. Every policy statement is based on some attributes [1]. This means that strength of statements depends on the values of the attributes. These attributes may not be the same as the data provider requires, but be slightly different. This means that the statement is fulfilled, but not entirely or even better than it is required.

Example 3: The data provider requires that "data must be deleted after 30 days" but the data consumer may guarantee only that "data will be deleted after 90 days" because the local law requires to keep the data for this period.

One solution is to search for a provider which can guarantee the policy statements as the client requires, but this solution is too idealistic in the most real-world situations. We propose an alternative solution: accept the data consumer which has the "closest" constrains on attributes to the desired ones. The data provider should specifies the strength of its statements depending on the attributes. We define a function str which is specific for each policy statement and determines the strength of a statement depending on the values of the corresponding attributes.

Definition 2: If A is a set of all possible attributes and P_d is a set of all desirable policy statements then by function str we mean the following mapping:

$$srt : P_d \times 2^A \mapsto \{perfect, high, medium, low, unacceptable\}$$

where by 2^A we mean a set of sets of attributes.

Note, that although the functions are defined for desirable policies they use the attributes taken from the corresponding guaranteed policy statements.

Together with usual levels (high, medium and low) we use two extremes: *perfect*, which means that a requirement is considered very well protected (almost perfect), and *unacceptable*, the strength level which the data provider cannot accept. We admit that the first level is rarely can be used because even the most robust requirement cannot give 100% protection. Nevertheless, we can assume that some attributes can make a policy statement almost perfect (e.g., encryption of data with a strong encryption standard has a very small probability to be broken). Unacceptable level is more useful and determines the border line that the data provider does not wish to cross by any circumstances.

Example 4: Continuing Example 3 for the policy statement "data must be deleted after N days" the data provider specifies that in case $N \leq 30$ the strength of the attribute is high and if $N \geq 60$ the strength is low. Actual deletion of data later than 180 days after the request about deletion is unacceptable. Thus in our example (with $N = 90$) calculated strength is "low".

In the paper we assume that every data consumer has a huge list of guaranteed policy statements and every data provider can find all its desirable policies in this list. Alternatively, a data provider can assign the lowest strength level (depending on the defined str function) to the policy statements which were not found in the guaranteed list. Note, that it is not required to define the function str for all possible results. Especially, perfect and unacceptable levels can be often omitted.

Figure 3 summarises the preparation procedures described above. First, ranks are assigned to desirable policy statements. Then, the desirable policy statements are mapped with the policy statements guaranteed by a data consumer. Finally, the strength of guaranteed requirements is computed.

[1]We use a simple approach for calculation of risk using qualitative values but the method may be performed in a quantitative way, though the quantitative way is much harder [9], [10]

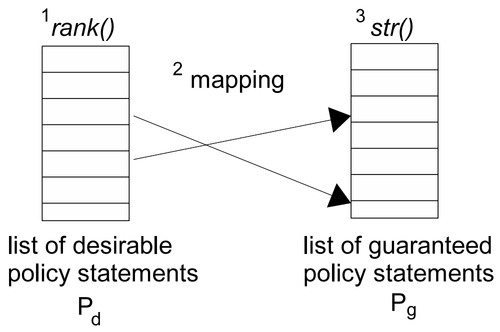
1 rank()
2 mapping
3 str()
list of desirable policy statements
P_d
list of guaranteed policy statements
P_g

Fig. 3. Preparation schema

Now we are ready to determine how well a concrete data consumer satisfies the requirements of the data provider. For this purpose we should compute the risk of each requirement to fail to fulfil its purpose. Usually risk is seen as a combination of three components: *Impact*, *Threat*, and *Vulnerability* [12]. *Impact* means the loss the data provider will face if a security breach occurs. *Threat* represents intention to brake the system when *Vulnerability* stands for possibility to do this. Computing a risk value various assessment methodologies (e.g., [12], [13]) use statistics to find the likelihood of breach occurrences. This likelihood can be seen as a product of *Threat* and *Vulnerability* components. The next step in computing a risk value (*R*) is to multiply the likelihood (L) by the amount of losses caused by one of such occurrences (*Impact*) (I):

$$R = L * I$$

We have to make a decision before the interaction starts and thus we cannot use likelihood like it is done in classical risk assessment methodologies. On the other hand, we have $rank$ which represents combination of *Impact* and *Threat* components. The exposure to the violation of a policy (*Vulnerability*) can be seen as a reverse value of the strength (function str) of the requirement. The logic here is that the more robust a policy is the less chance is that it will fail to fulfil its purpose. The calculation of the reverse value is straightforward: $perfect \mapsto unacceptable$, $high \mapsto low, medium \mapsto medium$ and viceversa[2].

Example 5: The data provider has rated the strength of the requirement "data must be deleted after 90 days" guaranteed by a possible data consumer as having 'low' strength. This means that there is a 'high' exposure for the data to be abused in this period.

Now for each requirement it is possible to compute the risk using a table similar to the Risk-Level Matrix used by NIST [12] extended with the two extremes (Table I). In Table I rows are the rank levels and columns are exposure levels. Risk is the value of the cell situated on the intersection of given rank and exposure levels.

[2]We renamed level 'perfect' with name 'no risk' (risk)

rank/exp	**perfect**	**low**	**medium**	**high**	**unacceptable**
low	no risk	low	low	low	unacceptable
medium	no risk	low	medium	medium	unacceptable
high	no risk	low	medium	high	unacceptable

TABLE I
QUALITATIVE CALCULATION OF RISKS

Definition 3: We define the $risk$ function as follows:

$$risk : \{unacceptable, high, medium, low, perfect\} \\ \times\{high, medium, low\} \mapsto \\ \{unacceptable, high, medium, low, no\ risk\}$$

In other words, if we would like to compute the risk for a policy $p_i \in P$ which requires attributes $a_{1,i}, ..., a_{n,i} \in A$ the function will be:

$$risk(p_i) = \overline{str}(p_i, a_{1,i}, ..., a_{n,i}) * rank(p_i)$$

where $\overline{str}$ is a reverse operation applied to the result of the function str.

The overall result of such calculation is a list of policy statements with assigned qualitative risk levels.

IV. Risk-based authorisation

Now we can return to the comparison of different services in order to select the one which satisfies policy statements in the best way. We employ the idea presented in [14] for comparison of multi-objective qualitative values. First of all, all providers which have at least one "unacceptable" risk are eliminated from the further consideration. Second, the numbers of high, medium, and low risks are summed up separately for each service provider. In other words, a data provider can assign the following tuple to each suitable data consumer:

Definition 4: Risk for a data consumer dc is a tuple: $R_{dc} = \langle h, m, l\rangle$, where h is a number of risks with high value, m - with medium risks, l - with low risks. We also can define the functions $HIGH$, $MEDIUM$ and LOW: $2^{RD} \mapsto N$ where RD is a risk domain $\{unacceptable, high, medium, low, no\ risk\}$ and N is a natural number domain. These functions simply return a number of high, medium and low risks taking a set of risks as an argument.

The data provider selects only the data consumers with the lowest number of high risk requirements. Among the selected data consumers only those who have less medium risks are taken. Finally, those who have less low risks are determined. These service providers satisfy desirable policy statements in the best way and the data provider must select one of these consumers to provide the data.

Example 6: Assume that we found three data consumers which got the following risks according to our computations: $\langle 3, 5, 7\rangle$, $\langle 4, 4, 8\rangle$, $\langle 3, 6, 6\rangle$. After the first check only the first and the third data consumers are left. The second check indicates that the first consumer has less medium risks. Thus the first data consumer should be selected.

V. Risk-based usage control

In the Section IV we considered granting authorisation to a data consumer which guarantees the most suitable protection for the data. Since relations between the partners may last for a long period (days, months, and even years) the attributes may change. UCON is based on the idea that the usage of data must be controlled also after authorisation. In this section we consider how a risk-based decision about further access can be made during usage of data.

Some policies can be enforced by a data provider when fulfillment of others should be observed. The enforced policies force the data consumer to behave only in a predefined way. Observable policies can be violated by the data consumer but the data provider monitors fulfillment of this requirements and can discover the abuse. Then the overall risk level is adjusted correspondingly.

UCON assumes that attributes may be simply changed (mutability of attributes) during access. These changes may affect both observable and enforced policies. We assume that if an enforced policy is affected by a change of attributes then the data consumer has to contact the data provider and ask him to adjust the enforcement controls. Monitoring of observable policies simply detects the change and notifies the data provider. About changes in other guaranteed policies (which cannot be neither enforced nor observed) the data consumer must simply notify the data provider itself.

Our idea is not to jump to a quick decision and revoke access to the data because one attribute became less strong than it has been agreed. We propose to look at the overall picture and only then make a decision.

First of all the data consumer should determine the current value of an attribute. Some attributes can be uniquely identified (e.g., used cryptographic technology) when others should be processed first. The data provider has to define how to determine the value of such attributes.

Example 7: After some time of interaction the data consumer notifies the data provider that from now on its data will be processed also by a specialised subsidiary situated in EU. Though this change violates one of the previously negotiated policy, i.e., "data must be processed only by the data consumer", this change is not considered by the data provider as a serious threat. The attribute (i.e., entities which process data) in this example is determined uniquely.

Example 8: Monitoring has shown that some data was removed 90 days after deletion as this was agreed, but some data was removed only after 120 days after deletion request. These delays were detected by the monitoring mechanisms and mean that the data consumer violates the negotiated policy: "completely remove data after 90 days from a delete request". The data provider decides to take the maximal time to complete removal as a real value of the attribute and this adjustment changes the risk of the data consumer. The attribute in this example may have different values and needs to be processed (find maximal value) before re-calculation of risk.

The changed attributes are used by the same function computing the strength of the requirement (see Section III). Then the current risk level is computed. If the risk is higher than some previously predefined threshold the access to the data is revoked. Otherwise, the data provider can still work with the data consumer.

The same idea can be used if we want to compare the current risk level of the current data consumer with other alternative providers. Note, that in this case we will compare the real (monitored) state of the current data consumer with guaranteed (not yet checked level) policy statements of other consumers.

VI. Service provider's side

Attributes of a service provider (data consumer) can also be improved. In other words, the data consumer may enforce new, more secure data management practices, install new security controls, change old encryption algorithms with new ones, etc. All these improvements of infrastructure mitigate the risk level of the data consumer and thus make its service more attractive to its clients.

In a simple scenario a data consumer may notice that its risk level according to preferences of a data provider is too high. If the data consumer does not want to loose its client it can improve its system in order to satisfy the data provider's requirements. Of course, the data consumer has to know the strength functions and threshold risk level used by the data provider. Since these functions are not secret we think that this assumption is reasonable.

Example 9: Assume that after some time of operation the attribute of the requirement "delete data after 90 days" has changed because of the latest local law amendment. Now the data can be deleted only after 120 days. This fact raises the risk of the data consumer and the client (data provider) thinks of changing the service provider. In order to decrease the high risk level the service provider decides to improve other attributes. The service provider decides to perform an external audit twice a year, but not just once as it was before.

Having limited security budget the data consumer can select only those security practices and techniques ("security controls" in the sequel) which improve its risk level best of all. First, we should find the security controls which reduce the number of high risks. Then, the rest money can be spent on the controls reducing medium risks. And finally we should consider low risks.

Definition 5: We define a function $change$ which transforms attributes if a security control (from a security control set SC) is installed as follows:

$$change : SC \times 2^A \mapsto 2^A$$

Now we can easily find how good is a security control cs_j, i.e., how many high risks become medium and low after its installation. Let $\hat{h}_{sc_j}$ be this difference, then it can be computed as follows:

$$\hat{h}_{cs_j} = HIGH(\{\overline{str}(p_i, a_{1,i}, ..., a_{n,i}) * rank(p_i) | \forall p_i \in P\}) -$$

$$HIGH(\{\overline{str}(p_i, a'_{1,i}, ..., a'_{n,i}) * rank(p_i) | \forall p_i \in P\})$$

where $a'_{k,i} \in A' = change(cs_j, A)$ and A is the initial set of all attributes ($a_{k,i} \in A$).

Also every security control has its cost c_{cs_j}. The overall security budget is W. Some security controls which must be implemented together in order to mitigate a risk we consider as one complex countermeasure with the summarised cost. Note, that if the atomic security controls are able to mitigate some risks not only as a part of a compound structure but by their own they must be also considered separately.

Now we can formalise the problem:

$$maximise \sum_{j=1}^{m} \hat{h}_{cs_j} * x_j$$

$$\sum_{j=1}^{m} c_{cs_j} * x_j < W$$

$$x_j = \{0, 1\} \quad j = 1, ..., m$$

$x_j = 1$ in the formula means that the corresponding security control (cs_j) have been selected, $x_j = 0$ otherwise.

This is a classical knapsack problem [15] which can be solved in pseudo-polynomial time, e.g., by dynamic programming. After selection of the security controls mitigating high risks the evaluation should be continued for medium risks with the rest money:

$$W_{medium} = W - \sum_{j=1}^{m} c_{cs_j} * x_j$$

Here we assumed that there were no unacceptable risks in the beginning because a service provider is able to know the desires of its clients only when it has agreements with them. And since the agreement exists we conclude that the clients have not found unacceptable risks in the guaranteed policies. If a service provider may compute risk functions for the client which have not selected the provider (e.g., by collecting wishes through a questionnaire of potential clients), then it has to eliminate as many unacceptable risks as possible first of all.

A service provider usually has more than one client. Thus the provider has to improve his system in order to fail as less agreements as possible. In order to solve this more complex problem we should simply consider policies for different clients separately computing the profit value:

$$\hat{h}_{cs_j} = \sum_{l=1}^{z} (HIGH(\{\overline{str}^l(p_i, a_{1,i}, ..., a_{n,i}) * rank^l(p_i) | \forall p_i \in P\}) -$$

$$HIGH(\{\overline{str}^l(p_i, a'_{1,i}, ..., a'_{n,i}) * rank^l(p_i) | \forall p_i \in P\}))$$

The assumption we have done for the calculation is that all installed together security controls do not conflict with each other.

VII. Related Work

Usage control can be seen as an evolution of access control. The main idea of usage control is that access to data can be revoked during usage (after granting access) of the data if some policies have been violated [1]. Thus usage control heavily depends on attributes which can change during usage of data (mutability of attributes) and, therefore, states that access decisions should be reconsidered if there were such changes (continuity of an access decision) [8].

Risk-based access control approaches are close to our work though access decisions in access control is usually made using one statement when we consider a number of policy statements which must be guaranteed during the whole time of data usage. Usual problem which authors try to solve is whether risk of possible abuse of policies by a user is less than the possible benefit from granting the access. For example, Dimmock et. al., [16] integrated information about risk and cost into OASIS policy language for reasoning about granting access. The authors of [17] presented an approach which decides whether to allow transactions on the difference between profit and risk. In contrast to our work the authors do not define how risk and profit should be computed, but focus on formalising information flow. Similarly, RAdAC [18], [19] is based on the idea that access should be granted taking into account risk and operational needs (can be considered as profit). Unfortunately, the available information about RAdAC does not contain many details. B. Aziz et. al., [20] considered different types of risks (operational, combinatorial and conflict of interests) and formalised usage of access control policies empowered with these risks in RBAC.

A close article to our work is [21]. The authors show how risks can be computed. Moreover, the authors used attributes (parameters) for such computations though, in contrast to our work, the attributes denote only possible context. We also used qualitative analysis which allows avoiding subjective values that are hard to find (e.g., probability of specific context). P. Chen et. al., [22] also presented their way to compute the risk that a subject will abuse an object. The authors focus only on levels of subject and object when we consider various policies defining how the object must be used.

From SOA perspective the work by N. Kokash and V. D'Andrea [23] is close to our idea. The authors also used risk of failure of a requirement to fulfil its purpose to select the most suitable web services. In our work we considered several policies (which can be considered as requirements in SOA) in order to compose risk explicitly. We also have shown how attributes can be used for computing risks and facilitate policies mapping. The AssessGrid project [24], [25] also is intended to empower selection of service provider with risk in a Grid. The project mainly focuses on timely delivery of results and considers risk only as a probability to satisfy this requirement [26], [27], [28].

VIII. Discussion

The analysis we propose in this paper is based on a simple model and can be easily conducted. The most difficult part of the analysis is determination of functions str and $rank$. Though both these functions must be subjectively defined by the data provider, in practice, preferred values of parameters and importance of policies should be defined anyway. The method only shows how to aggregate the assessments of individual policies in order to make a decision. Naturally, the complexity of function str grows rapidly with increase of amount of attributes for a policy. On the other hand, in many practical cases policies have only a few attributes. Also many complex policies can be simply broken down into smaller ones.

Our approach is focused on analysis of attributes and decision making (grant access to the data or do not). Thus we left behind the question about collection of the attribute values. We assume that the values which our analysis is based on are trustworthy. Thus we do not consider this, also important for UCON, part of the problem about monitoring of values and their delivery to the access decision point.

IX. Conclusion and Future Work

In this work we empowered the usage control in SOA with risk. This is the first work which uses risk in usage control to the best of our knowledge. Risk allows efficiently map desirable and guaranteed policies and select the service provider which guarantees the usage of data in the most desirable way. Moreover, we have shown that there is no need to revoke access to data if some attributes have become worse than they were before. Our attribute-based risk analysis allows performing a comprehensive assessment during usage of data. Using our basic idea we also indicated how a service provider can effectively improve its infrastructure in order to provide the most suitable service to its customers.

There are a number of ways how the main idea presented in the paper can be elaborated. We can improve the basic approach proposed in this paper by detailed analysis and formalisation of possible usage control policies in order to facilitate the mapping of desirable and guaranteed policies. Another way for improvements could be to consider benefits from granting access to a specific data consumer versus risk connected with giving control over the data to it. In this paper we did not use trust considering selection of service providers. Employing trust will help to rely not only on values provided by the data consumer, but also have some information about correctness of these values. Finally, so far we compared service providers according to how "bad they are", selecting the less risky one. We can look at the problem from the other ("bright") side and consider the provider which is more secure, i.e., which has the strongest protection for the most important policies.

References

[1] J. Park and R. Sandhu, "Towards usage control models: beyond traditional access control," in *Proceedings of the 7th ACM Symposium on Access Control Models and Technologies*. New York, NY, USA: ACM, 2002, pp. 57–64.

[2] A. Pretschner, M. Hilty, and D. Basin, "Distributed usage control," *Communications of the ACM*, vol. 49, no. 9, pp. 39–44, 2006.

[3] M. Hilty, A. Pretschner, C. Schaefer, and T. Walter, "Enforcement for Usage Control: A System Model and a Policy Language for Distributed Usage Control," DoCoMo, Tech. Rep. I-ST-20, December 2006.

[4] A. Pretschner, F. Massacci, and M. Hilty, "Usage control in service-oriented architectures," in *Proceedings of the 4th International Conference on Trust, Privacy & Security in Digital Business.*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2007, vol. 4657.

[5] F. Martinelli, P. Mori, and A. Vaccarelli, "Towards continuous usage control on grid computational services," in *Proceedings of the International Conference on Autonomic and Autonomous Systems and on Networking and Services*, 2005.

[6] B. Aziz, A. Arenas, F. Martinelli, I. Matteucci, and P. Mori, "Controlling usage in business process workflows through fine-grained security policies," in *TrustBus-08: Proceedings of the 5th international conference on Trust, Privacy and Security in Digital Business*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 100–117.

[7] S. A. Butler, "Security attribute evaluation method: a cost-benefit approach," in *Proceedings of the 24th International Conference on Software Engineering (ICSE'02)*. ACM Press, 2002, pp. 232–240.

[8] M. Alam, X. Zhang, M. Nauman, T. Ali, and J.-P. Seifert, "Model-based behavioral attestation," in *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*. New York, NY, USA: ACM, 2008, pp. 175–184.

[9] F. Cohen, "Managing network security - part 5: Risk management or risk analysis," *Network Security*, vol. 1997, no. 4, pp. 15–19, 1997.

[10] A. Stewart, "On risk: perception and direction," *Computers & Security*, vol. 23, no. 5, pp. 362–370, 2004.

[11] I. Brandic, D. Music, and S. Dustdar, "Service mediation and negotiation bootstrapping as first achievements towards self-adaptable grid and cloud services," in *Proceedings of the 6th international conference industry session on Grids meets autonomic computing*. New York, NY, USA: ACM, 2009, pp. 1–8.

[12] G. Stoneburner, A. Goguen, and A. Feringa, "Risk management guide for information technology systems," National Institute of Standards and Technology, Tech. Rep. 800-30, 2001, available via http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf on 13/05/2009.

[13] C. J. Alberts and A. J. Dorofee, "OCTAVE Criteria," CERT, Tech. Rep. CMU/SEI-2001-TR-016, December 2001.

[14] S. Venkataraman and W. Harrison, "Prioritization of threats using the k/m algebra," in *Proceedings of Workshop on Software Security Assurance Tools, Techniques, and Metrics*, 2005, pp. 90–95.

[15] D. Pisinger, "Algorithms for knapsack problems," University of Cophenhagen, Tech. Rep. DK-2100, 1995.

[16] N. Dimmock, A. Belokosztolszki, D. Eyers, J. Bacon, and K. Moody, "Using trust and risk in role-based access control policies," in *Proceedings of the 9th ACM Symposium on Access Control Models and Technologies*. New York, NY, USA: ACM, 2004, pp. 156–162.

[17] L. Zhang, A. Brodsky, and S. Jajodia, "Toward information sharing: Benefit and risk access control (barac)," in *Proceedings of the 7th International Workshop on Policies for Distributed Systems and Networks*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 45–53.

[18] R. W. McGraw, "Risk-adaptable access control (RAdAC)," available via http://csrc.nist.gov/news_events/privilege-management-workshop/radac-Paper0001.pdf on 16/08/09.

[19] R. Choudhary, "A policy based architecture for nsa radac model," in *Proceedings of the Workshop on Information Assurance*, 2005.

[20] A. B. Aziz, A. S. Foley, A. J. Herbert, and A. G. Swart, "Reconfiguring role based access control policies using risk semantics," *Journal of High Speed Networks*, vol. 15, no. 3, pp. 261–273, 2006.

[21] N. N. Diep, L. X. Hung, Y. Zhung, S. Lee, Y.-K. Lee, and H. Lee, "Enforcing access control using risk assessment," in *ECUMN '07: Proceedings of the Fourth European Conference on Universal Multiservice Networks*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 419–424.

[22] P.-C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger, "Fuzzy multi-level security: An experiment on quantified risk-adaptive access control," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 222–230.

[23] N. Kokash and V. D'Andrea, "Evaluating quality of web services: A risk-driven approach," in *Proceedings of the 10th Conference on Business Information Systems*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2007, vol. 4439.

[24] K. Djemame, I. Gourlay, J. Padgett, G. Birkenheuer, M. Hovestadt, O. Kao, and K. Voss, "Introducing risk management into the grid," in *Proceedings of the Second IEEE International Conference on e-Science and Grid Computing*. Washington, DC, USA: IEEE Computer Society, 2006, p. 28.

[25] M. Hovestadt, O. Kao, and K. Vob, "The first step of introducing risk management for prepossessing slas," in *Proceedings of the IEEE International Conference on Services Computing*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 36–43.

[26] D. Battré, K. Djemame, O. Kao, and K. Voss, "Gaining user's trust by publishing failure probabilities," in *Proceedings of the First International Workshop on Security, Trust and Privacy in Grid Systems*, 2007, pp. 193–198.

[27] A. Keller, K. Voss, D. Battre, M. Hovestadt, and O. Kao, "Quality assurance of grid service provisioning by risk aware managing of resource failures," in *Proceedings of the Third International Conference on Risks and Security of Internet and Systems*, 2008, pp. 149–157.

[28] D. Battré, K. Djemame, I. Gourlay, M. Hovestadt, O. Kao, J. Padgett, K. Voss, and D. Warneke, "Assessgrid strategies for provider ranking mechanisms in risk—aware grid systems," in *Proceedings of the 5th international workshop on Grid Economics and Business Models*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 226–233.