D.3.2 – State of the art of purpose-based, usage control approaches

Georges Nassopoulos, Patricia Serrano-Alvarado, Pascal Molli, Emmanuel Desmontils

D.3.2 – State of the art of purpose-based, usage control approaches

ANR-13-INFR-0003
socioplug.univ-nantes.fr

Georges Nassopoulos, Patricia Serrano Alvarado, Pascal Molli, Emmanuel Desmontils

[1]*LINA UMR 6241 / Universite de Nantes.*
Email contact: Patricia.Serrano-Alvarado@univ-nantes.fr

---

1 Context and objectifs

1.1 Problem statement

Federation of plugs makes users real owners of their data i.e., there is no a third party provider that hosts personal data with a service agreement. This allows end-users to define their own policies for their data and services. Therefore, data can be shared with personal policies specifying what can be done and what is forbidden, without depending on a collaboration provider. Consequently, it prevents information asymmetry problems and abusive usage of personal data.

Our work focuses on the problem of:
"How to ensure usage control in community-regulated federations"?

This problem is related to usage control (UCON) [PS02b, Zha06], a model that emerged recently as a solution for modern applications to preserve access control in open, distributed, heterogeneous, and network-connected environments. The particularities of UCON are continuity of access decision evaluation and mutability of several properties (i.e., attributes of subjects or objects) considered during the access decision.

We consider that the evaluation of access decision should take into account policies of involved subjects (not only providers but also identified subjects) and eventual conflicts between concurrent users should be resolved.

In this open, untrusted and federated network, we need another asset to guarantee a continuous access control. This asset is *provenance* [CCT09, MCF+11, CAB+14], which generally is identified as a meta-data that records the ancestry, derivation, or history of some object, explaining it's current state. We can use this information/meta-data for a more compact perception of the context, in which access may be granted or denied.

1.2 Objective

Our objective is to ensure usage control in community-regulated federations and for this we are oriented towards:

1) Proposing an adequate access control model which respects the below criteria, inspired by works in collaborative systems [TAPH05] and Web Based Social Networks (WBSN) [CFP09]:

- **(i) Distribution:** must be applied and enforced into a federated network.
- **(ii) Expressiveness:** should be expressive enough to specify access rights efficiently based on varied information (e.g., roles, context), for a diversity of objects and levels of granularity.

(iii) Scalability and Concurrency: data would be used and transferred between a large number of users, that may eventually try to access these data concurrently.

(iv) Performance and Cost minimization: concerning multi user collaboration tasks, these features must be kept within acceptable (and subjective) bounds fixed by the federation's participants.

(v) Continuity of Control: control should be enforced before, during and after access is granted.

(vi) Dynamicity and Mutability: should be possible to specify and change policies, depending on the environment and mutability of attributes.

(vii) High level authority design: must allow high level specification of access rights, for better managing the increased complexity of (tuple-level) fine-grained policies.

(viii) FOAF architecture and trust: take into account that data are shared, based on both direct and indirect trusted relationships between nodes. This follows the *Friend-Of-A-Friend principle (FOAF)*† [KGG+06], where a policy is defined not only by the type but also the maximum depth of a (ranked trusted) relationship.

2) Collecting and analysing provenance: we want to exploit provenance, in order to enrich access control decision with the following characteristics, as summarized in [SPG05]:

(a) Data quality and trust: provenance can be used to estimate data quality and data reliability based on the trust score of source data and transformations (which is based on owners trust scores). It can also provide proof statements on data derivation.

(b) Audit Trail and accountability: used to trace the audit trail of data, establish the copyright and ownership of data, enable its citation, determine resource usage, detect errors in data generation and determine liability in case of erroneous data.

(c) Replication recipes and antisymmetry: detailed provenance information can allow repetition of data derivation, help maintain its currency and be a recipe foe replication (which is important, as we want to emancipate from central collaboration providers).

This objective can be faced through eager or optimistic approaches. The former should ensure usage control and guarantee usage policies are always preserved, the latter should allow data owners to verify that their usage policies are preserved and to take adequate measures when misusage is detected. As it is well known eager mechanisms are more expensive, we thus think that a verification-based approach will better scale in federations. we propose to adapt distributed usage control approaches to federation of plugs and explore how usage control approaches can be used to address the important problem of right to oblivion.

Next sections survey existing access control mechanisms and provenance followed by the positioning of these approaches over our context.

First a rapid overview is given of various access control models, in Section 2. Next, we present the provenance domain, which is organized in two parts: "workflow" and "data" provenance in Section 3. Finally we compare these approaches in Section 4, based on characteristics we mention in Section 1.

2 Access control overview

This section illustrates the state of the art concerning access control, in order to identify the aspects interesting to our work.

2.1 Traditional access control models

1) Mandatory Access Control (MAC):

Initially proposed for military systems, MAC is at the core of the requirements for trusted computing. The most popular model for MAC is the *Bell and LaPadula* model [BL73]. This approach is based on security classification (CLS) of objects and clearance of subjects (CLR), for which MAC requires at least

† http://xmlns.com/foaf/spec/

a partial order between them (CLS and CRS elements, respectively). To simplify, we have the following levels of sensitivity, which form a total order:

U unclassified < *C confidential* < *S secret* < *T top secret*

For subjects (users), the sensitivity represents the level of access; for objects (data item), it represents the level of protection. A subject can access an object, if the subject's CLR dominates an object's CLS.

In the Bell and LaPadula security model, there are 2 basic restrictions:

(a) (No read-up): a subject at a given security level may not read an object at a higher security level. Example: a soldier (with CLR "unclassified") may read the program of a military exercise (with CLS "unclassified") but cannot read the technical report (with CLS "Secret").

(b) (No write-down): a subject at a given security level must not write to any object at a lower security level. Example: a general (with CLR "Top Secret") may read the technical report of a military exercise (with CLS "Secret") but cannot write on the exercise program (with CLS "unclassified").

2) Discretionary Access Control (DAC)

In contrast to MAC, DAC approach is discretionary, meaning that authorized users have the **capacity of delegation**: transfer their rights directly or indirectly to other subjects and thus having control over original data, not copies. Access is granted based on the identity of subjects and/or groups to which they belong. Typically DAC permits authorized users or administrators to define an *Access Control List (ACL)* on a matrix of users and resources, containing *Access Control (ACE)* which defines each user's privileges for a set of resources.

3)Role-Based Access Control (RBAC) [SCFY96]

RBAC was introduced as a *refinement of DAC*, providing alternative roles that capture the type of rights and the degree of discretion that a group of users may have. This feature eases the management process of assessing authority, resulting to a more scalable approach. First, each user in the system is assigned to one or more roles **(role assignment)**. Then each of these roles, which correspond to a set of permissions and privileges, is controlled for authorization of the requesting user **(role authorization)**. Finally, the user can exercise a requested action on a subject, only if it's permitted for one or more of the user's active role **(permission authorization)**. In practice, RBAC can implement also MAC by simulating the Lattice-Based Access Control (LBAC) based on the concepts of role hierarchy and constraint **(RBAC as a superset of LBAC)**.

DAC, MAC and RBAC have been proved to function efficiently in closed and trusted environments, each one with its advantages and constraints: MAC is simple and scalable but too rigid; DAC is more fine-grained but not scalable; RBAC is more manageable and scalable than DAC but "role engineering" is costly and tedious [Coy95]. None of these models was conceived to function in an open and untrusted environment.

2.2 The Usage Control model (UCON$_{ABC}$)

Towards the UCONABC family [LMM10] In order to be able authorizing access in an open and untrusted environment, two major technologies appeared in the mid 90's. First, **Trust Management (TM)** [BFL96, AG07] which defines security policies and trust relationships between peers but treats only with static (not mutable) entities. Second, **Digital Rights Management (DRM)** [SBVW95, Sch99] which deals with intellectual property rights protection and unfortunately bring about a limited set of usage scenarios to protect only multimedia objects.

At this point, **UCON** [PS02b, Zha06] emerged as a new access control model able to enforce fine-grained, flexible, persistent and continuous access and usage of digital resources. UCON enhances traditional access control models in two novel aspects: mutability of attributes and continuity of an access decision. It is capable to cover DAC, MAC, RBAC, TM, DRM models and goes beyond them: If during the ongoing access attributes are changed and the security policy is not satisfied any more, the UCON authorization system revokes granted rights and terminates the resource usage.

2.2.1 UCONABC components and state transition system

In UCON, security policy statements and access decisions are determined by three factors: Authorizations, oBligations and Conditions (thus called $UCON_{ABC}$), as we see in Fig. 1.

1) Authorization: predicates put constrains on subject's and/or object's attributes (e.g. subject's name must be "John" and object's type ".doc").

2) Obligations: actions that are required to be performed by a subject before, during or after the usage of an object (e.g. a subject must sign a license browsing a web-page).

3) Conditions: environment restrictions that are required to be valid before or during the usage (e.g. an object is available on working hours only).

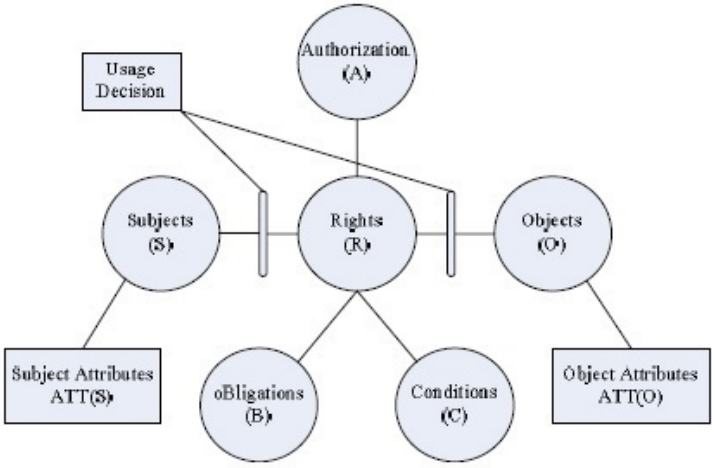

Figure 1: UCON decision model. [Zha06]

The UCON transmission state system defines the following usage process (Fig. 2):

i) Initial state: means that access request is not generated.

ii) Requesting check state: indicates that the access has been generated and is waiting for the system's usage decision.

iii) Denied state: refers to the state where the system has denied access.

iv) Accessing state: means that the system has permitted access and the subject is accessing the object immediately after.

v) Ongoing check state: refers to the continuity of control, in which the system checks for conditions and obligations fulfilment.

vi) End and Revoked states: the termination of the access done either when the system revokes the access after it has been granted in requesting state or it is ended normally by the user.

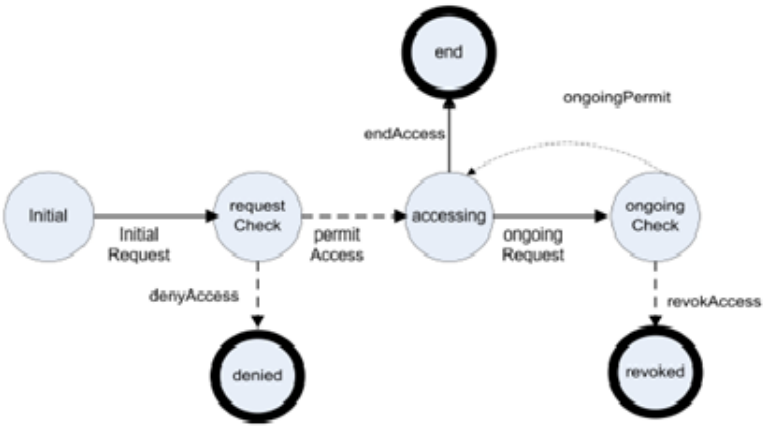

Figure 2: UCON transmission state system. [KZB+08]

Nota Bene: Later in [KZB+08], the state *Ongoing check* was added in UCON to cover post obligations, and we think this can be useful to implement right to oblivion.

Some work has been done, in the aim of enforcing UCON in distributed systems. Pretschner et al. [PHB06] introduced the Distributed UCON (DUCON) meta-model to express the concepts of "enforcement" and "observability" along with the appropriate infrastructure. In [HPB+07a] Obligation Specification Language (OSL) was introduced as a policy language for DUCON, able to cover post-obligations. In [HPB+07b] authors tackle the problem of policy evolution in usage control when data is re-distributed. Finally in [KP13] and [KPPK11] some architectures for usage enforcement over distributed networks and Web Based Social Networks (WBSN) are deployed, respectively.

2.3 Other access control models

In this section, we overview some (of the many) Access Control (AC) models, that usually are enforced complementary with the methods illustrated previously. We will focus more on those models that exploit provenance as a history of data usage and derivation, to enrich access control as motivated in Section 1.

2.3.1 AC based on dissemination/flow control:

a) Originator Control (ORCON) [Pro81]: brought forward to restrict the distribution of documents in paper world, requires recipients to gain originator's approval for re-dissemination of objects but lacks of performance and scalability: each time that some data is to be disseminated, the approval of the owner must be granted again. Also used in UCON [PS02a], it's interesting in our context as data may be shared between peers through direct and indirect relationships.

b) Dissemination Control (DCON) [Gra89, MMN90]: extends ORCON with dissemination control before and after access is granted, by respecting the initial requirements of the owner (without his re-approval). It is used also as an extension of DRM, and consequently used in UCON. In [LY06] authors proposed a Dynamic Multi-Policy Dissemination Control model (DMDCON) to express the dynamic and multi-policy nature existing in reality, which is also one of our goals (see Section 1).

2.3.2 AC based on history and provenance of data usage:

The following models collect provenance and log data usage and/or derivation to enrich access control and represent it in the form of *workflow* or *data* provenance, as we will see in Section 3.

a) History Based Access Control (HBAC) [BN04]: access is granted based on the action and request history of the subjects. In this context, the main motivation is to differentiate the "good behaviour" of subjects, from misusage . This is close to provenance of data usage, and thus connected with the next model.

b) Provenance Based Access Control (PBAC) [PNS12]: uses provenance meta-data for controlling access to data themselves. In this work, data transactions and provenance information as a Direct Acyclic Graph (DAG). The model uses the edges of this graph, representing transactions (called "causal dependencies") between objects, processes but also users to enforce access control. PBAC is based on *workflow provenance* and formalized with *Open Provenance Model* (as we will see in the next Section 3). Some first attempts have been made to use this model to enhance UCON [Bie13].

c) Provenance Access Control (PAC) [BSS08, CKKT11]: concerns how access to provenance of data should be controlled, as provenance data might be more sensitive than the data itself. PAC implementation in [CKKT11], captures provenance in the same way than PBAC in [PNS12], using workflow provenance and the Open Provenace Model. In this context, provenance is secured by exploiting causal dependencies between objects and assessing authority to use them in combination with an other access control model (e.g., RBAC).

PBAC and PAC models are complementary to each other in a way that PBAC can be used to control access to provenance data and PAC can be used to elevate trustworthiness of provenance data. Furthermore, they both require mechanisms to capture, store and retrieve provenance data and PAC is essential for this, as it identifies how provenance data should be structured and retrieved.

d) Tuple Based Access Control (TBAC) [TLT14]: set of tuple based access control models enforcing DCON over relational data. TBAC is MAC-based, as it proposes an approach for "information flow control": the use of some data doesn't mean that it can be re-disseminated without control. This model is similar also to PBAC, as it uses provenance to control access to data, with the difference that it relies on *data provenance* (using *provenance semirings*, which we will also see in the next Section) and not on *workflow*.

Our goal is the conception and enforcement of an access control model that exploits provenance to ensure privacy and data protection but that will also protect provenance itself, as motivated in Section 1. For this, in the next Section we will see the provenance framework to capture and manipulate provenance.

3 The provenance framework

Provenance generally is identified as meta-data that records the ancestry, derivation, or history of some information, explaining it's current state. In the next paragraphs we will see the two main categories of provenance models: *"worflow"* and *"data"* provenance, but there are identified also other types as we may see in [CAB+14], depending on the captured granularity level.

3.1 Workflow provenance

Scientific Workflow Management Systems (SWfMS)‡ describe transactions involving subjects, objects, and the corresponding actions describing the interaction between them (for scientists to prototype and execute in silico experiments). The applicability of such systems, concerns a much wider variety of programming constructs than databases including: concurrency, procedures, service calls and queries to external databases.

Workflow provenance aims to capture a complete description of evaluation of a workflow ("procedural data processing"), which is crucial to verification in SWfMS computation. In this description, each module (set of operations) is a *black-box*, so that each output of the module depends on all its inputs, due to its complexity (coarse-grained dependencies).

In this context, provenance is modelled and record as a Directed Acyclic Graph (DAG). The **Open Provenance Model (OPM)** [MCF+11] has been developed as a consensus exchange format for representing provenance graphs by a workflow system. Without causality dependency as semantics foundation, it is hard to utilize transaction flows and associated information; and this is the reason that OPM was created.

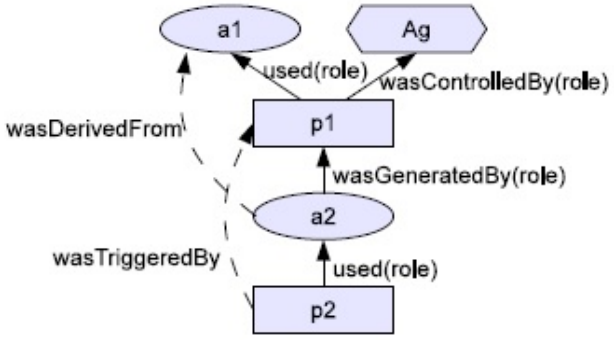

Figure 3: OPM Causality Dependencies. [PNS12]

As you can see in Fig 3, OPM main components consist of artifacts, processes, and agents. Five main dependencies between two components are defined as *"used"* (process on artifact), *"wasGeneratedBy"* (artifact on process), *"wasControlledBy"* (process on agent), *"wasDerivedFrom"* (artifact on artifact), and *"wasTriggeredBy"* (process on process).

An extended survey and taxonomy of existing provenance systems, such as Taverna, Kepler, etc., is presented in [dCCM09]. In the "workflow provenance" context, PBAC and PAC models have been proposed to use provenance to enrich access control decision and to protect provenance information it self, respectively (see Section 2).

3.2 Data provenance

We use the term *data* or *database* provenance, broadly to refer to a description of the origins of a piece of data and the process by which it arrived in a database.

The most common forms of database provenance are:

‡ http://www.wfmc.org/

1) where provenance (cell level): describe relationships between data in the source and in the output, for example, by explaining where output data came from in the input [BKT01];

2) why provenance (tuple level): show inputs that explain why an output record was produced [BKT01];

3) how provenance (tuple level): describe in detail how an output record was produced [GKT07], which includes "why" provenance.

Besides being interested in understanding the behaviour of queries, these forms of provenance have been used in the study of classical database problems, such as view update and the expressiveness of update languages. More recently, they have also been used in the study of annotation propagation and updates across peer-to-peer systems [CCT09].

In next Section we overview *how* provenance.

3.3 "How provenance": the provenance semiring

Green et al. [GKT07] have introduced provenance semiring as the appropriate structure to capture provenance of tuples. Every tuple of the database is annotated with an element of a *commutative provenance K-semiring* $(K, \oplus, \otimes, 0, 1)$ and annotations are propagated through query evaluation. This framework extends positive Relational Algebra (RA^+), in the sense that classical RA^+ is extended to non-boolean valued relation (as k-semirings are an abstract annotation that includes boolean and go beyond).

(A) Semiring utility in a nutshell: semirings annotations capture the way in which the result tuple has been derived from input tuples. Semiring addition $\oplus$ corresponds to alternative derivation of a tuple, thus, the union of two relations corresponds to adding up the annotations of tuples appearing in both relations. Similarly, multiplication $\otimes$ corresponds to joint derivation, thus, a tuple appearing in the result of a join will be annotated with the product of the annotations of the two joined tuples.

(B) Semiring algebraic axioms: The RA identities of a commutative semiring[§] bellow, must be respected when data (and their annotations) are combined:

1) (K, $\oplus$, 0) is a commutative monoid ($\oplus$ associated to $\cup$ and π)

associative: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

commutative: $a \oplus b = b \oplus a$

unity element: $a \oplus 0 = a = 0 \oplus a$

2) (K, $\otimes$, 1) is a commutative monoid ($\otimes$ associated to $\bowtie$)

associative: $(a \otimes b) \otimes c = a \otimes (b \otimes c)$

commutative: $a \otimes b = b \otimes a$

absorbing element: $0 \otimes a = 0 = a \otimes 0$

3) ($\otimes$ is distributive over $\oplus$)

$a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

$(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$

(C) The universal semiring[¶] (Fig. 4). Green et al. in [GKT07] introduced also the universal commutative semiring N[X] and proved that it generalize all semirings, as we may see in Fig. 4. This is why how provenance is also identified as *implicit* provenance: N[X]-semiring captures the abstract structure and properties of query operators and can, thus, be used for various annotation computations explicitly using the *evaluation homomorphism* $N[x] \rightarrow K_i$ (from the universal semiring to the desired semiring).

(D) Orhcestra[‖] **data provenance** (Fig. 5). Initially the provenance semiring was introduced to capture provenance information in Collaborative Data Sharing Systems (CDSS) [KGIT13, TI10]. Karvounarakis in

[§] A semiring is commutative when (K, $\otimes$, 1) is also commutative

[¶] www.cs.ucdavis.edu/ green/slides/spring09-job-talk.pptx

[‖] https://dbappserv.cis.upenn.edu/home/?q=node/8

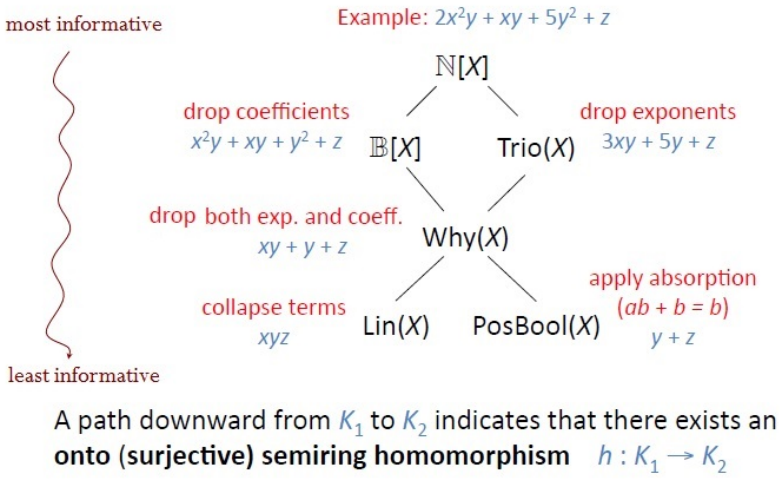

Figure 4: Todd J. Green thesis defence.

[Kar09], extended semirings with a set of unary functions, one for each mapping (query), and showed that they satisfy the fundamental theorem, as well as that the corresponding extension of provenance polynomials is the most general such structure.

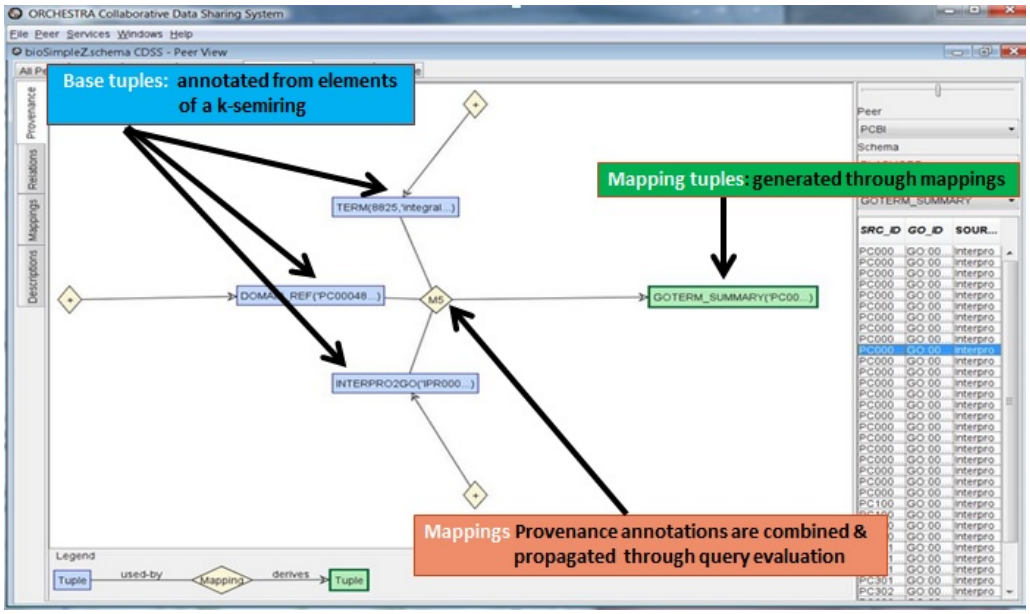

Figure 5: ORCHESTRA provenance graph.

In Fig. 5, we see how ORCHESTRA captures and records provenance**. This system represent provenance as a graph, where we can see with blue colour the *base* tuples (inserted manually into the Data Base), with pink the *mappings* (queries) through which annotations are propagated and with green the *mapping* tuples (generated through mappings).

(E) Related work on Semirings: Authors of [GKT07] but also other researching groups, have studied various aspects of semiring usability and applications such as: Karvounarakis et al. [KIT10] define ProQL as appropriate query language for provenance in ORCHESTRA; Geerts et al. [GKCF13] create the SPARQL minus semiring (spm-semiring) for RDF graphs; Foster et al. [FGT08] extends the semring framework to XML and introduce the "confidentiality" semring; but also many other research fields, as they are summarized in [KG12].

Next, we will see a cross comparison between the access control mechanisms and provenance models, respectively, based on the criteria we presented in Section 1.

4 Quick analysis of state of the art

As motivated in Section 1, we illustrate at this point an evaluation of the presented access control models (inspirited by works in collaborative systems [TAPH05] and Web Based Social Networks (WBSN) [CFP09]), but also a comparison between provenance models and their applications (as they are identified in [SPG05]).

First in Table 1, we start with a comparison†† between the access control presented presented in Section 2, according to the criteria of Section 1:

	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
MAC [BL73], DAC [Lam74], RBAC [SCFY96]	-	*	-	-	-	-	+	-
TM [BFL96, AG07], DRM [SBVW95, Sch99]	+	+	-	-	-	-	+	+
$UCON_{ABC}$ [PS02b, Zha06]	*	+	*	-	+	+	+	+
ORCON [Pro81], DCON [Gra89, MMN90]	*	+	-	-	+	-	+	*
HBAC [BN04], PBAC [PNS12]	+	-	-	-	-	-	+	*
PAC [BSS08, NXB+09, CKKT11]	-	-	-	-	-	-	+	*
TBAC [TLT14]	+	-	-	-	+	-	+	*

Table 1: Cross AC models comparison, for a social-collaborative environment

In this Table, we observe that $UCON_{ABC}$ covers the majority of our objectives and enriched with provenance information it would be a good candidate to integrate in our work.

In Table 2, we can see a comparison between applications of the two basic provenance models, as presented in Section 3, also based on characteristics of Section 1.

	(a)	(b)	(c)
Data provenance [GKT07]	+	X	*
Workflow provenance [MCF+11]	*	+	+

Table 2: Workflow Vs Data provenance

Comparing workflow and data provenance, we conclude that the first can be used for more applications, with the constraints of space complexity (due to usage log size) and lack of fine-grained granularity (as it captures procedural dependencies).

** https://dbappserv.cis.upenn.edu/home/

†† '+', means that the system treats this case, - means the system does not treat this case and '*' means the model partially treats this case

References

[AG07] Donovan Artz and Yolanda Gil. A survey of trust in computer science and the Semantic Web. *J. Web Sem.*, 5(2):58–71, 2007.

[BFL96] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized Trust Management. In *IEEE Symposium on Security and Privacy*, pages 164–173, 1996.

[Bie13] Christoph Bier. How Usage Control and Provenance Tracking Get Together - A Data Protection Perspective. In *IEEE Symposium on Security and Privacy Workshops*, pages 13–17, 2013.

[BKT01] Peter Buneman, Sanjeev Khanna, and Wang Chiew Tan. Why and Where: A Characterization of Data Provenance. In *ICDT*, pages 316–330, 2001.

[BL73] D Elliott Bell and Leonard J LaPadula. Secure computer systems: Mathematical foundations. Technical report, DTIC Document, 1973.

[BN04] Anindya Banerjee and David A. Naumann. History-Based Access Control and Secure Information Flow. In *CASSIS*, pages 27–48, 2004.

[BSS08] Uri Braun, Avraham Shinnar, and Margo I. Seltzer. Securing Provenance. In *HotSec*, 2008.

[CAB+14] Lucian Carata, Sherif Akoush, Nikilesh Balakrishnan, Thomas Bytheway, Ripduman Sohan, Margo Seltzer, and Andy Hopper. A primer on provenance. *Commun. ACM*, 57(5):52–60, 2014.

[CCT09] James Cheney, Laura Chiticariu, and Wang Chiew Tan. Provenance in Databases: Why, How, and Where. *Foundations and Trends in Databases*, 1(4):379–474, 2009.

[CFP09] Barbara Carminati, Elena Ferrari, and Andrea Perego. Enforcing access control in Web-based social networks. *ACM Trans. Inf. Syst. Secur.*, 13(1), 2009.

[CKKT11] Tyrone Cadenhead, Vaibhav Khadilkar, Murat Kantarcioglu, and Bhavani M. Thuraisingham. A language for provenance access control. In *CODASPY*, pages 133–144, 2011.

[Coy95] Edward J. Coyne. Role engineering. In *ACM Workshop on Role-Based Access Control*, 1995.

[dCCM09] Sérgio Manuel Serra da Cruz, Maria Luiza Machado Campos, and Marta Mattoso. Towards a Taxonomy of Provenance in Scientific Workflow Management Systems. In *SERVICES I*, pages 259–266, 2009.

[FGT08] J. Nathan Foster, Todd J. Green, and Val Tannen. Annotated XML: queries and provenance. In *PODS*, pages 271–280, 2008.

[GKCF13] Floris Geerts, Grigoris Karvounarakis, Vassilis Christophides, and Irini Fundulaki. Algebraic structures for capturing the provenance of SPARQL queries. In *ICDT*, pages 153–164, 2013.

[GKT07] Todd J. Green, Gregory Karvounarakis, and Val Tannen. Provenance semirings. In *PODS*, pages 31–40, 2007.

[Gra89] Richard Graubart. On the need for a third form of access control. In *Proceedings of the 12th National Computer Security Conference*, pages 296–304, 1989.

[HPB+07a] Manuel Hilty, Alexander Pretschner, David A. Basin, Christian Schaefer, and Thomas Walter. A Policy Language for Distributed Usage Control. In *ESORICS*, pages 531–546, 2007.

[HPB+07b] Manuel Hilty, Alexander Pretschner, David A. Basin, Christian Schaefer, and Thomas Walter. A Policy Language for Distributed Usage Control. In *ESORICS*, pages 531–546, 2007.

[Kar09] Grigorios Karvounarakis. *Provenance in collaborative data sharing*. PhD thesis, University of Pennsylvania, 2009.

[KG12] Grigoris Karvounarakis and Todd J. Green. Semiring-annotated data: queries and provenance? *SIGMOD Record*, 41(3):5–14, 2012.

[KGG+06] Sebastian Ryszard Kruk, Slawomir Grzonkowski, Adam Gzella, Tomasz Woroniecki, and Hee-Chul Choi. D-FOAF: Distributed Identity Management with Access Rights Delegation. In *ASWC*, pages 140–154, 2006.

[KGIT13] Grigoris Karvounarakis, Todd J. Green, Zachary G. Ives, and Val Tannen. Collaborative data sharing via update exchange and provenance. *ACM Trans. Database Syst.*, 38(3):19, 2013.

[KIT10] Grigoris Karvounarakis, Zachary G. Ives, and Val Tannen. Querying data provenance. In *SIGMOD Conference*, pages 951–962, 2010.

[KP13] Florian Kelbert and Alexander Pretschner. Data usage control enforcement in distributed systems. In *CODASPY*, pages 71–82, 2013.

[KPPK11] Prachi Kumari, Alexander Pretschner, Jonas Peschla, and Jens-Michael Kuhn. Distributed data usage control for web applications: a social network implementation. In *CODASPY*, pages 85–96, 2011.

[KZB+08] Basel Katt, Xinwen Zhang, Ruth Breu, Michael Hafner, and Jean-Pierre Seifert. A general obligation model and continuity: enhanced policy enforcement engine for usage control. In *SACMAT*, pages 123–132, 2008.

[Lam74] Butler W. Lampson. Protection. *Operating Systems Review*, 8(1):18–24, 1974.

[LMM10] Aliaksandr Lazouski, Fabio Martinelli, and Paolo Mori. Usage control in computer security: A survey. *Computer Science Review*, 4(2):81–99, 2010.

[LY06] Zude Li and Xiaojun Ye. Towards a dynamic multi-policy dissemination control model: (DMDCON). *SIGMOD Record*, 35(1):33–38, 2006.

[MCF+11] Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul T. Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, Beth Plale, Yogesh Simmhan, Eric G. Stephan, and Jan Van den Bussche. The Open Provenance Model core specification (v1.1). *Future Generation Comp. Syst.*, 27(6):743–756, 2011.

[MMN90] Catherine Jensen McCollum, Judith R Messing, and L Notargiacomo. Beyond the pale of MAC and DAC-defining new forms of access control. In *Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on*, pages 190–200. IEEE, 1990.

[NXB+09] Qun Ni, Shouhuai Xu, Elisa Bertino, Ravi S. Sandhu, and Weili Han. An Access Control Language for a General Provenance Model. In *Secure Data Management*, pages 68–88, 2009.

[PHB06] Alexander Pretschner, Manuel Hilty, and David A. Basin. Distributed usage control. *Commun. ACM*, 49(9):39–44, 2006.

[PNS12] Jaehong Park, Dang Nguyen, and Ravi S. Sandhu. A Provenance-Based Access Control model. In *PST*, pages 137–144, 2012.

[Pro81] Intelligence Resource Program. Director of Central Intelligence, Control of Dissemination of Intelligence Information, Directive No. 1/7. May 4 1981.

[PS02a] Jaehong Park and Ravi S. Sandhu. Originator Control in Usage Control. In *POLICY*, pages 60–66, 2002.

[PS02b] Jaehong Park and Ravi S. Sandhu. Towards usage control models: beyond traditional access control. In *SACMAT*, pages 57–64, 2002.

[SBVW95] Olin Sibert, David Bernstein, and David Van Wie. Digibox: A self-protecting container for information commerce. In *Proceedings of the first USENIX workshop on electronic commerce, New York, NY*, pages 1–13, 1995.

[SCFY96] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2):38–47, 1996.

[Sch99] Paul B Schneck. Persistent access control to prevent piracy of digital information. *Proceedings of the IEEE*, 87(7):1239–1250, 1999.

[SPG05] Yogesh Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance in e-science. *SIGMOD Record*, 34(3):31–36, 2005.

[TAPH05] William J. Tolone, Gail-Joon Ahn, Tanusree Pai, and Seng-Phil Hong. Access control in collaborative systems. *ACM Comput. Surv.*, 37(1):29–41, 2005.

[TI10] Nicholas E. Taylor and Zachary G. Ives. Reliable storage and querying for collaborative data sharing systems. In *ICDE*, pages 40–51, 2010.

[TLT14] Romuald Thion, Francois Lesueur, and Meriam Ben-Ghorbel Talbi. Tuple-Based Access Control: a Provenance-Based Dissemination Control Model for Relational Data. Technical report, INSA, Lyon, 2014.

[Zha06] Xinwen Zhang. *Formal Model and Analysis of Usage Control*. PhD thesis, George Mason University, Fairfax, VA, USA, 2006. AAI3221391.