Adnosco: trace user data for the user

Nadia Bennani* — **Fabien Duchateau**** — **Előd Egyed-Zsigmond*** — **Philippe Lamarre***

* *Université de Lyon*
CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France
** *Université de Lyon*
CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622 France
firstname.lastname@liris.cnrs.fr

ABSTRACT.
The development of the web has seen the explosion of web forms as a mean for the user to provide information to web applications in the context of both personal and professional activities. Currently, user personal data are stored and managed at companies' side. That makes the user dependent on the corresponding services while granting her a very passive role. We aim to demonstrate that if correctly modeled, user traces provide many benefits for both the user and services. In this paper, we propose a trace model including a new web form qualification model and a user-friendly solution that improves web users productivity by providing semantics-based tools. One of our goals is to provide trace analysis methods that enable answering questions like : "Where have I read something about Inforsid on the web?"

RÉSUMÉ. *Avec la démocratisation du web, la quantité d'informations transmises par des utilisateurs vers des sites web a explosé. Actuellement ces données sont stockées sur les serveurs des entreprises et une fois transmises, deviennent inaccessibles pour leurs auteurs. Nous nous proposons de démontrer qu'en traçant les actions de l'utilisateur selon des modèles de traces bien définispermettant ainsi aux utilisateurs de conserver leurs informations, nous pou- vons faire bénéficier de ces informations à la fois aux utilisateurs mais également aux entreprises destinataires de ces informations. Dans ce papier, nous présentons un modèle de traces incluant la qualification sémantique de pages web et une solution à base d'extension de navigateur qui propose des outils de traçage. Un de nos objectifs est de proposer des méthodes d'analyse permettant de répondre à des questions du type : "Sur quelle page web ai-je obtenu des informations à propos d'Inforsid ?".*

KEYWORDS: *trace for the user, vrm, web trace model*

MOTS-CLÉS : *traçer pour l'utilisateur, vrm, modèle de trace web*

1. Introduction

Advancements in computer technology have largely reduced the hardware costs especially for storage. Currently, almost all web sites have the capacity to record much information for and about their users. Furthermore, on one hand, many tools (CMS, CRM, data mining) have been developed to allow sites to obtain, store, access, manage and exploit data provided by their users. On the other hand, a user has no tools to manage her data submitted online. As a result, the access to information is increasingly unbalanced since, paradoxically, the more computers capacities increase, the more the user loses control over data she submits on the web. To illustrate this point, let us think about: who can make an exhaustive list of information explicitly transmitted using web forms, when they have been transmitted, to who, and for which purpose?
Without contesting the right for sites to store data, it seems natural and highly complementary, to enable the user to record and structure her own data. In addition, the user may take advantage of migrating some data to her personal applications. For example, in the context of an online purchase, the transaction amount can be transmitted to her personal account manager and an event can be set in her calendar to remind the delivery date. Yet, this is not currently possible.
Such arguments seem to reach the point of view developed by the Berkman Center For Internet & Society at Harvard University when they present the notion of Vendor Relationship Management (*VRM*) (Havard, 2007) which is the customer-side counterpart of the more known Customer Relationship Management (Wikipedia, 2013). While the expected benefit of a CRM are turned to the vendor, the VRM focuses on individuals. Five main properties have been identified (Havard, 2007). Our objective is to go further, proposing a tool compliant with those which are related to our context: "Customers must be the points of integration for their own data"; "Customers must have control of data they generate and gather. This means they must be able to share data selectively and voluntarily"; and "Customers must be free to express their demands and intentions outside of any company's control".

To move towards this goal, in this paper, we present our first proposal, ($Adnosco$) a tool which enables to model, trace, store, search and manage submitted data. Due to lack of space, we do not intend to present all the potential uses for stored user traces. We will focus rather on the benefits of storing, at the client side, data submitted to web sites. Local storage of data submitted through web form fields raises several problems starting with their acquisition. Some operational solutions, developed in other contexts, already exist (e.g., Lazarus (Interclue, 2011), Dashlane (Dashlane, 2013)), and we look at them as proofs of concept. We mainly focus on using the user traces for the input help. Indeed, **completion** and **pre filling** are functionalities which have already shown their ability to induce significant productivity gains in professional contexts and which can rely on user data.

In many cases, it is interesting for the user to reuse the already submitted data. That is the spirit in which many browsers propose values. But they limit their proposals to values previously typed in the same field. Our first syntactical approach proposes the

completion and pre filling functionalities, but it goes a step ahead by exploiting stored personal data rather than just data already filled in by all users. However, when filling in a web form for the first time, data can only come from other web forms and in such case, the syntactic approach has a weak accuracy due to the high heterogeneity of web forms. To improve precision, we propose to manage heterogeneity by introducing a semantic based-approach. Lastly, considering that a user must often fill in different web forms to achieve a higher goal, we introduce the concept of "activity" that makes concrete an aspect of the "user's context". The intuition is simple: the data involved in an activity are often repetitive. For example, a trip planning activity goes through different web sites for travel, accomodation, etc. Again, it is important to keep in mind that these assistance tools are devoted to the user who is $infine$ the one that can judge the coherence of the data to be submitted. Hence any proposed tool should not be intrusive. It is up to the applications and services in concern to enforce their rules in order to obtain consistent data.

This paper is outlined as follows: section 2 gives a formal definition of web forms and presents a motivating scenario. Section 3 shows a generic definition of completion and pre filling functionalities. Sections 4, 5 and 6 show how syntactic, semantic and activity based approaches deal with user's data. Section 7 presents a theoretical evaluation of our proposal. Finally related works are discussed in Section 8 before concluding.

2. Background

Since the web forms are central to our study in this paper, it is important to define them formally. We abstract from presentation and language (HTML version, embedded code...) as well as from all technical points to focus only on information related to our issue.

Definition 1 (Web form)
A web form *wf is a tuple* $\langle uri, fields, tt \rangle$ *where*

– uri *is an URI*

– $fields(wf)$ *is a finite non empty set of* n *fields* $\{f_{wf,1}, f_{wf,2}, ..., f_{wf,n}\}$
A field is a triple $\langle name, type, value \rangle$.

– tt *is the Transaction Time (i.e. the submission time stamp).*
tt *is set to* $null$ *until the form is submitted.*

3. Proposal General Sketch: Functionalities

Our general concern is about the loss of control of a user over the information she submits on the web. This paper does not claim to address this problem in its generality. Rather we focus on a first step which is to acquire these data, to store them, to structure them and to exploit them in such a way that the user obtains clear

advantages. A beneficial side effect could be to raise individual user awareness about transmitted data potential.

Assuming these data acquired, we propose to help the user to fill in web forms via three functionalities to enhance the user experience and to enable significant productivity gain: restoring values of a web form to get it exactly as it was submitted (given a submission date); proposing possible completions to the user while she is trying to fill in a form field; and, pre-filling fields when the user has not yet entered any value for the form field and for which the server has not proposed default values. These functionalities are well known in many other applications, but, even considering recent advances of our browsers, when not completely absent, they are available in very marginal situations. Having access to the whole user data, $Adnosco$ can offer an extended and powerful pre-filling and completion functionalities. Let us briefly present how we consider these functionalities in $Adnosco$. Then we will detail the different methods for the completion functionality in the three next sections.

Restoring values. Restoring webform field values as they were submitted at a certain date is not a so complicated task as far as filled web forms are stored. The Dashlane (Dashlane, 2013) and in some measure the Autofill Form Firefox plugins (Interclue, 2011) propose such a functionality restricting to recent submissions which is already really helpful.

Completion and pre filling are more complex functionalities. Following this intuition, we propose a set of methods each of them producing a set of relevant values ordered according to a partial pre-order encoding their relative relevance. Each method answers the same question in different ways. The general question is "*in the context of the considered web form (already filled values, already typed value in the current field...) which values to propose to the user?*". A complementary question is, "*how to order these results?*". The proposed answers are stored in a simple triple $\langle$methodName, $ValueSet$, $\leq \rangle$. We will then obtain as many of these triples as there are proposed methods. Methods can be ordered according to their assumed accuracy. Thereby, the global results of all invoked methods are ranked in the list L: from the one assumed to be the most accurate, to the last. This order can be determined a priori by the application or set up by the user.

The next section is devoted to different methods to obtain relevant values and associated pre-orders. We mainly explore three possible ways.

1 - *Syntax-based approach* helps the user filling in the web form fields based on syntactic criteria. In this case, proposed values have to be selected with respect to syntactic consideration only, i.e., same field or same type. 2 - *Semantics-based approach* brings semantic notions within the picture to obtain more accurate results. 3 - *Activity-based tool* proposes to take into account some user's context introducing the notion of "activity". The intuition is to know what the user is doing to better understand how its current web form filling is related to previous ones. at the end of section 6 , we will illustrate thanks to our scenario, the calculation of the list L and how it is exploited to display the list of possibilities when trying to fill in a field.

4. Extracting Values and Relevance Order based on Syntax

We propose two different syntactic methods to extract values relevant to a field of a web form. The first is simply to consider only what the user has already submitted through the same field during a previous submission of the same web form while the second increases the scope by searching all available values in fields having the same type, regardless the web form where they appear.

4.1. *Location based approach: same web form, same field*

4.1.1. *The value set*

can be obtained looking only at values already filled in the same field. More formally, the value set can be defined as: $ValueSet^{SF}_{syntactic}(wf, f, c, W)$ where wf is the web form under concern, f is the field under focus, c is the actual field value ($null$ if none is present), and W is the set of web forms to take into account. Except in some particular cases, usually, this last is set to WF (the set of all known web forms).

4.1.2. *Associated preorder candidates*

Many preorders are natural candidates to qualify the relevance of obtained values:

– $\leq_{sim}$ is a preorder over values which is obtained considering similarity between equivalent web forms.

– $\leq_{freq}$ is a preorder over values which is obtained considering the frequency of a term.

– $\leq_{trans}$ is a more simple pre-order which just considers the transaction times.

– $\leq_{nat}$ is the simplest among proposed pre-orders. It does not pay any attention to the web form but it focuses on the natural order of values according to their type (alphabetical, numerical, etc.).

To close this consideration on orders, one could be interested to combine them. It is possible, and for example, $\leq_{trans-sim}$ denotes the composition where two values are first ordered using $\leq_{sim}$, and, in case of equality are ordered using $\leq_{trans}$. According to the resulting order, the top corresponds to the most recent value among those which appears in most similar web forms.

4.2. *Type based approach*

4.2.1. *The value set*

is obtained considering only the values of the same type than the type of the concerned field, in previous submitted web forms. The number of presented values is higher than for the syntactic same field method

4.2.2. *Associated pre-order candidates*

Excepted for the similarity approach which requires equivalent web forms, previously introduced notions can be used here. Conversely to $\leq_{trans}$ and $\leq_{nat}$) which can be used without any modification, due to the fact that a value may appear more than one time in a web form, $\leq_{freq}$ which is based on frequency, has to be adapted.

5. Extracting Values and Relevance Order based on Semantics

As we saw in the previous section, syntactic methods reach their limits very quickly. For the first one, its research field is too limited to enable to find the desired value. On the contrary, for the second one, it enlarges considerably the result set as it brings a very (too) large number of values, no order being able to bring out relevant ones. Clearly, the type of data is not a sufficiently accurate criteria to distinguish between values. Better structuring is needed to improve performance. Unfortunately, unlike the site that produces the form, a user does not have the information she needs about the structure of the data contained in a form. To break the deadlock, we propose to bring semantics (ontologies) into the picture. Obtaining ontologies and semantic qualification will also be a problem. There are many possibilities. They can be built by the user, but this is a huge work for a single person. They can be provided by the sites as they provide CSS style sheets, but we are far from that. Or they can be built and shared by users communities and associations. Whatever, before thinking about how to obtain them, we have to define them and to evaluate how interesting they are, for our objective.

5.1. *Semantic qualification of web forms*

To semantically qualify a web form, one can think to link web form fields to concept properties. However, in presence of multiple instances of the same concept within the same web form, this technique does not allow to distinguish them. For instance, in figure 1 there are information about two addresses with two names, counties, ... A simple syntax based assistance cannot distinguish between the two field sets concerning the two addresses. To obtain a semantic qualification with a higher structuring power, we propose to introduce the notion of *materialized concepts*. A *materialized concept* links a group of fields to a concept of an ontology, associating each field to a property of this concept.

5.1.1. *Formal definition*

Definition 2 (Materialized Concept) *A* materialized concept associated to a web form wf *(definition 1) is a quadruple* $\langle name, concept, wf, Corr \rangle$ *where*

– $name$ *is the name of the materialized concept which is a simple string. It should be unique for a web form.*

– *$concept$ is a concept of an ontology.*
The set of properties associated to that concept are noted $Props(concept)$.

– *wf is the web form to which the materialized concept is associated.*

– *$Corr$ is a set of triples $\langle fieldName, op, property\rangle$ precizing how fields of the web form materialize the associated $concept$:*

- $fieldName$ is the name of a field of wf,

- op is an operation which enables to deduce the field value considering the value of a property (in this paper, for the sake of simplicity, we consider only the identity (=) symmetric operation).

- $property$ is a property of the concept $concept$.

Unsurprisingly, a semantic qualification of a web form may involve more than one materialized concept.

Definition 3 (Semantic qualification of a web form) *A* semantic qualification sq of a web form wf*(definition 1) is a triple $\langle name, wf, MC\rangle$, where*

– *$name$ is a unique name of the semantic qualification.*

– *$wf \in WF$ is a web form concerned by the semantical qualification.*

– *MC is a set of materialized concepts associated to the web form wf, i.e. $\forall mc \in MC$, $mc.wf = wf$.*

The semantic qualification of a web form is done independently of the embedded values. In other terms, if a semantic qualification is done for a web form wf then it applies to any other equivalent web form.

Notations

– $SQ(wf)$ denotes the set of all known semantic qualifications of the web form wf, i.e. $SQ(wf) = \{sq : sq.wf = wf\}$.

– SQ denotes the set of all known semantic qualifications, i.e. $SQ = \bigcup_{wf \in WF} QS(wf)$.

5.2. *Using semantical qualification to extract value set*

Let wf be the web form under concern, $f \in wf.fields$ the field having the focus, and c the current value of the field.
$ValueSet_{Sem}(wf, f, sq, c, W)$ is the set of elements of the format $qs'.wf'[f']$ [1], such that:

1. reminder: in this paper, for the sake of simplicity, we only consider equality operation.

– f is semantically associated to a property of a materialized concept mc.
$\exists mc \in sq.MC$, $\exists al \in mc.Corr$: $al.fieldName = f.name$, and

– according to some semantic qualification sq', there exists fields semantically associated to the same property of the same concept.
$\exists sq' \in SQ$, $\exists mc' \in sq'.MC$, $mc'.concept = mc.concept$ and $\exists al' \in mc'.Corr$, $al'.property = al.property$, and

– among fields so qualified, we are interested into those which start with the value c
$\exists wf' \in EQ(sq'.wf) \cap W$: $((wf'[al'.fieldName]$ starts with $c)$ or $(c = null))$.

5.3. *Associated preorder candidates*

The resulting set can be ordered considering many criteria. $\leq_{freq}$, $\leq_{trans}$ and $\leq_{nat}$ are here again good candidates.

5.4. *Illustration scenario*

To illustrate Adnosco syntactic and semantic assistants efficiency, let us present the following scenario. John Smith is living in Lyon. He decides to offer a gift to his daughter Alice who is married and lives in Lille. John is used to buy his gifts on Internet. He connects to the *pc21.fr* website and fills in the main web form with his personal information, his address in Lyon considered as the billing address, the personal information and address information (considered as the delivery address) for his daughter. John has just downloaded Adnosco. He decides to experiment it and defines four materialized concepts framed respectively in red (Customer), orange (InvoiceAddress), blue (DeliveryAddress) and cyan (Recipient), one for each subset of fields described below. He also links semantically his personal information (red frame) with his address information(orange frame). He does the same link between her daughter personal information and address (see figure 1).

Later, John travels to Lille. For his return travel, he plans to take a train that arrives at 23:00 to Lyon. He then decides to book a taxi to get home from the railway station. To do this, he goes to the *taxis-lyonnais.fr* web site and fills in it. John never visited this site but the site's webmaster has linked semantically the fields *name (Nom ou Société)* and *e-mail* to the ontological concept *person* and the field *telephone* to the *address* ontological concept as it is bound to the physical address of somebody. The two sections *Departure address (Adresse de départ)* and *Arrival address* (figure 3) have been also qualified semantically and linked to the concept *address* using two distinct materialized concepts. John types 's' in the *name* field. As shown on figure 2, Adnosco displays in the context menu syntactic then semantic choices. The names *Smith* and *Snoopy* are proposed in the case of syntactic completion as they are both stored in the Adnosco data storage as possible values for the *name* field of the *pc21*

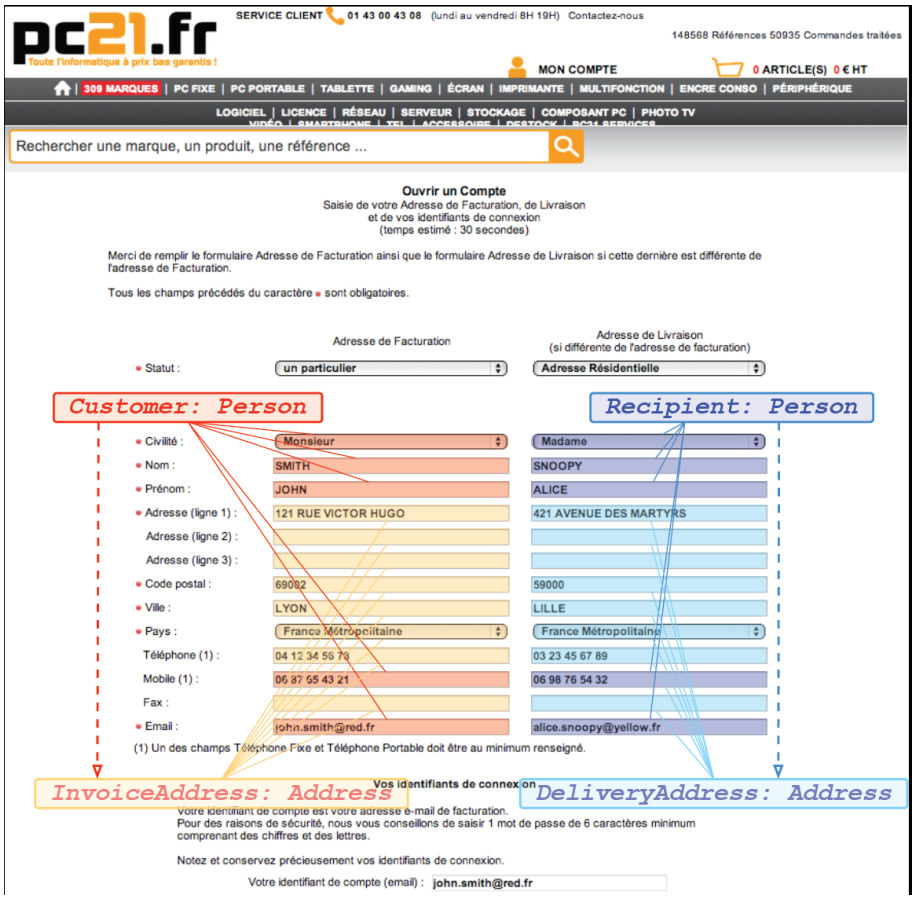

Figure 1: Materialized concepts on a webpage (pc21.fr)

web form. They are also proposed by the semantic assistant as they are the two values associated to the attribute *name* of the concept *person*. John chooses his name in the semantic set of choices. Immediately, the field *e-mail* is filled automatically by John's e-mail as in the storage the name and the e-mail are associated to the same materialized concept. Additionnally selecting *Smith* as the user name, implies the automatic fill in of the *telephone* field. In fact, as the *telephone* field is linked semantically to the same materialized concept as for the *address* fields in the *pc21.fr* web form, Adnosco proposes the phone number stored previously for John Smith when he filled the *pc21.fr* web form. Finally when John tries to fill in the *city (*ville)* field in the *Arrival address* section (see figure 3, the semantic assistance proposes 'Lyon' and 'Lille' as both are city names. The syntactic assistant doesn't give any proposal as John is visiting the *taxi-lyonnais.fr* web form for the first time.

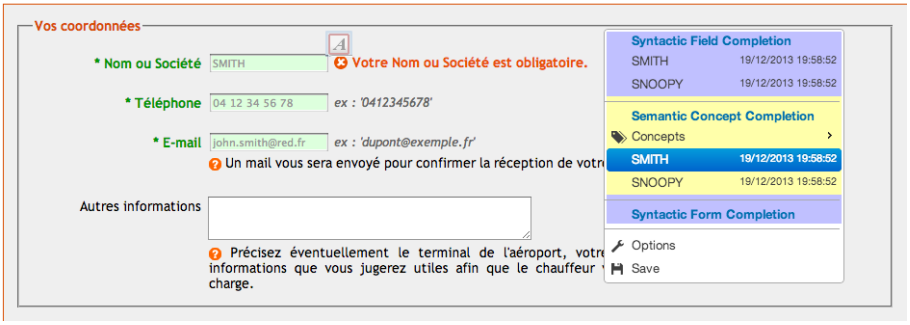

Figure 2: Illustration of Semantic assistance based on semantic qualification dependencies (taxi-lyonnais.fr). Adnosco is called for the field with the "A" icon. The green fields are the automatic semantics based completion proposals corresponding to the selected value in the menu.

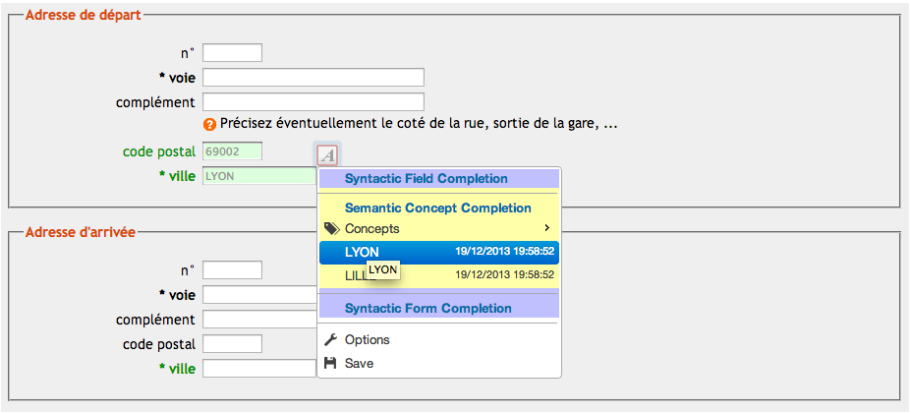

Figure 3: Illustration of Semantic assistance based on semantic qualification (taxi-lyonnais.fr)

6. Extracting Values and Relevance Order considering Activity

The previous section shows that semantics is helpful to propose relevant values to the user. We introduce the activity notion to make possible to cross data from one web form to another as far as they are part of the same activity, and this without any concern about the provider to which they have been uploaded nor about time from last connected data upload.

Intuitively, an activity polls many forms for a specific purpose. The concerned forms may have been designed specially for this purpose (ex.: set of web forms of

a flight booking site), or more interestingly have been developed independently (ex. flight booking site + car rental site + hotel site) and used together in a dynamic manner. To formalize the information overlap between forms, they are aggregated within an identified activity which corresponds to a particular context of use. For this purpose, we rely on materialized concepts which have been introduced to semantically qualify web forms.

Definition 4 *Activity*
An activity A *is a tuple* $\langle name, W_A, Q, C\rangle$ *where*

– $name$ is the name of the activity. We assume it to be unique (i.e. there is no different activities sharing the same name).

– W_A is the set of web form gathered in the activity $name$.

– Q is a set of semantic qualifications such that there is at most one semantic qualification per web form
$\forall(sq_1, sq_2) \in SQ^2$, if $sq_1.wf = sq_2.wf$ then $sq_1 = sq_2$.

– C is a set of correspondences between materialized concepts present in the semantic qualifications that belong to Q.
A correspondence between materialized concepts is defined as a tuple $\langle sq_1, mc_1, sq_2, mc_2\rangle$ such that

- *$\forall i \in \{1, 2\}$, $sq_i \in Q$*
- *$\forall i \in \{1, 2\}$, the materialized concept $mc_i \in sq_i.mc$*
- *$mc_1.concept = mc_2.concept$.*

This condition could be relaxed considering mapping between ontologies, but here it is out of scope of this paper. For the sake of simplicity, we restrict ourselves to consider only concepts of the same ontology.

Correspondance obtained by transitivity are automatically added.

$(wf, f) \simeq_{ad} (wf', f')$ denotes the fact that in the context of an activity defined by ad, the fields f and f' belonging respectively to the web forms wf and wf' are related to the equivalent materialized concept according to the activity ad. The interpretation of this correspondence is that, normally, the materialized concepts mc_1 and mc_2 should lead to instances with the same property values, i.e. identical values for their corresponding fields in the web form instances. The user still has the ability to waive this rule. For this he simply enter the values she wants, by ignoring those proposed by our system. We don't want to impose anything to the user, letting him fully responsible about data she communicates. The problem of checking integrity constraints is left to the sites.

Definition 5 (Activity instance)
An activity instance *is a a tuple* $\langle name, ad, WF_{ai}, st, ct\rangle$ *where*

– $name$ is the name of this instance, assumed to be unique,

– *ad is the activity on which this activity is based,*

– *WF_{ai} is a set of web forms involved within this instance activity.*

– *st is the starting time of the activity instance, and*

– *ct is the closing time of the activity instance.*

It is interesting to note that in an activity instance all web forms present in its associated definition do not have to be present and reversely, some web forms may be embedded into the activity by the user even if they are not semantically qualified.

6.1. *Extracting value sets with respect to an activity*

Here the objective is to extract values strongly related to the current activity. By definition, they are semantically related and so already proposed by the peviously presented semantic approach, but they are much less numerous. To be more precise, their number does not depend on all uploaded web forms but only on those belonging to the activity instance. We also take advantage of expressed correspondences over materialized concepts. The extracted value according to these rules set is noted $ValueSet_{Act}(wf, f, c, ai)$ where wf is the web form under concern, f is the field under focus, c is the actual field value ($null$ if none is present) and ai is the current activity instance. This method selects the values of fields linked to f through materialized concepts and activity correspondences, and whose values are compatible with c. More formally,
$ValueSet_{Act}(wf, f, c, ai)$=$\{wf'[f'] : wf' \in ai.WF_{ai}$ and $wf.f \simeq_{ad} wf'$ and $wf'[f']$ starts with $c\}$.

7. Evaluation

In this section, we evaluate the time gain in filling web forms when using $Adnosco$. Let us consider the following situation. A user has to fill in 3 web forms in order to book a journey. Let's say the first web form is about 2 traveler persons and contains 6 fields (Name, Surname,City)*2, the second webform is about travel information, containing 8 fields ((Name, Surname)*2, Outbound Dearture and Arrival date and Inbound Dearture and Arrival date) and the 3^{rd} one is about the car rental containing 5 fields (Name, Surname, Departure Date, Arrival Date, Car type). We also consider that the Name and Surname fields from the first 2 webforms are in correspondence through an Activity instance as well as the first Name and Surname field from the second webform with the Name and Surname of the third one. Without any assistance she has to fill the $6+8+5 = 19$ fields in the three web forms. Considering 0,4 seconds needed per character [2], 5 characters in average per word and 2 words in average per field, that would take at least 76 seconds. With a syntactic completion assistance after

2. Words per Minute, `http://en.wikipedia.org/wiki/Words_per_minute`

typing in the first characters, the system provides the correct value. This reduces to 4 the average character number to type per field and thus give a theoretical period of 30,4s. The semantics based assistance increases the precision of the recommended values and thus decreases the number of characters to type in a field to 2 giving 15,2 seconds of filling time. With an activity based pre filling, the time needed to fill the first web form remains the same. For the second web form, only the dates will be to precise (4 fields instead of 8), as for the third web form, the only field to fill in will be the $Cartype$. The total theoretical time necessary to fill in the three web forms will be reduced to $6 \times 2 \times 0,4 + 4 \times 2 \times 0,4 + 2 \times 0,4 = 8,8s$. This represents a theoretic gain of 88%.

To generalize, we estimate this gain through calculations. First we announce a set of hypothesis:

– the average length in characters of a word is LC (5 in the previous example)

– the average word count in a web form field is WC (2 in the previous example)

– the average number of character to type before getting a correct completion is TLC (4 in the previous example)

– the percentage of correct pre-filling over the set of web forms in a given activity is CPF (70% in the previous example)

We can say that if we have N web forms to fill in in a given activity, with n_i fields each ($i = 1..N$) and t is the average time to fill in a field, we need $\sum_{i=1}^{N} (LC * WC * n_i * t)$ time to complete the N web forms. Adding the assistance, this time is reduced to $\sum_{i=1}^{N} (TLC * (1 - CPF) * n_i * t)$. Without semantic assistance $TLC = LC * WC$ and without activity based assistance, CPF is 0.
The estimation of parameter values obtained on our simple example have to be verified nevertheless through experimentations on real data sets considering different activities.

8. Related Work

This section covers two domains : **applications for managing personal information** and **the alignment of data sources**.

There are several fields that tackle the management of user data. One of them is gathered around the online identity community and places the end-user at the center. They relay all communication between identity providers and service providers through the user's client (Bramhall *et al.*, 2007) (Cameron, 2005) (Marc Goodner, 2008) enabling people to have and employ a collection of digital identities. The Information Card metaphor is implemented by Identity Selectors like Windows CardSpace (Cameron, 2005) (Nanda, 2007) and the Higgins project (Higgins, 2007). An Identity Selector system generally provides the user with an interface to create and manage personal information cards. These works mainly provide solutions to avoid unsuper-

vised spreading of user data, but don't help her track and reuse information filled in web forms.

Another category of works includes applications that follow user actions inside web browsers, such as the Firefox plug-ins: $CoScripter, Lazarus, AutofillForms$ and $PrivacyDashboard$. While $CoScripter$ translates user actions in plain text, $Lazarus$ stores web form data and enables to refill the form as it was submitted at a given date. $AutofillForms$ is announced to be the closest one to $Adnosco$ but actually it doesn't work and has very little documentation. $PrivacyDashboard$ provides a control interface to check what kind of information is sent to which website. The $Collusion$ and $Moluti$ plug-ins enable to analyze surf surveillance and history, while (Worlfram|Alpha, 2014) provides Facebook data analysis.

Concerning the webform semantic qualification, the alignment task, also known as matching, has been studied for many decades (Batini *et al.*, 1986). Traditionnally, the structured data sources involved in alignment can be ontologies (Euzenat et Shvaiko, 2007), schemas (Bellahsene *et al.*, 2011) , or entities (Talburt, 2011). Alignment tools usually combine different similarity measures (e.g., instance-based, terminological, lexical, constraint-based) applied to the elements of the data sources (e.g., concepts, properties, instances). In ontology alignment, researchers compete during the annual OAEI challenge using various datasets to demonstrate the effectiveness and performance of their tools (Euzenat *et al.*, 2011).

A few works have focused on matching unstructured data sources, such as web forms. The SMB approach deals with the matching of two web forms from the same domain (converted in the OWL format) (Marie et Gal, 2008). In a similar fashion, the UIUC repository collects query interfaces to help understanding the modelling and integration of web databases (UIU, 2003) . Zhang et al have used the UIUC collection to discover a hidden syntax from web forms (Zhang *et al.*, 2004). At this point we are tending more towards ontologies proposed on $schema.org$ to align the webforms with.

Although this paper mainly presents the foundations of $Adnosco$, our tool is also designed for discovering semantic links between a web form and an ontology using similarity metrics. Contrary to all these alignment approaches, $Adnosco$ does not perform any alignment between two structured data sources (e.g., ontologies, schemas) or two unstructured data sources (e.g., web forms). The additional semantic layer proposed in our approach includes materialized concepts, which bridge the gap between a structured data source (an ontology) and an unstructured one (a web form). Besides, a materialized concept takes into account the instances of a web form, so that related data are stored together and can be proposed later with effectiveness. The syntactic and semantic assistants extend comparable solutions (Dashlane, 2013 ; MIT, 2013) to values issued from other websites and to more precise and rich propositions based on webform semantic qualification and the use of materialized concepts to handle multiple concept instances on the same web form.

9. Conclusion

In this paper, we have introduced $Adnosco$, a **user-centric personal data tracer and manager** that allows the user to store her own data submitted online through web forms. $Adnosco$ is also able to store the trace model including the organization to which the information has been transmitted, transaction timestamp and the mean of transmission (web form or any other data transmission). Besides, $Adnosco$ can be considered as a non-repudiation mean in case of lost web form submission. This paper mainly focuses on one of the advantages of $Adnosco$: its ability to assist efficiently the user in **filling in forms** on her navigator. To this end, when the user attempts to fill in a value, data is extracted from $Adnosco$ repository and proposed to her, in an order that corresponds to her chosen configuration. Three methods to extract data and their possible data orders has been proposed: syntactic, semantic and activity based. The **syntax-based** method proposed values corresponds to all data beginning with the same characters than the current filled in field, while the **semantic-based** method extracted data corresponds to data that is semantically qualified similarly than the current field; finally, the **activity-based** extraction is more selective as it limits the proposals to previous data filled in in the forms that belong to the same activity or more selectively to the same instance of some activity. Besides, activity-based assistance is empowered thanks to the novel notion of materialized concepts and their established correspondences, that *in fine*, precises the set of proposed values.

In the future, we plan to extend field correspondences to other operation types to increase the expressiveness of $Adnosco$ and facilitate the **discovery of semantic links** between data. Furthermore, we are working on an automatic tool to help users discover materialized concepts between an ontology and a web form. We are exploring two methods to fulfill this goal. The former matches the elements of the new web form directly with the existing materialized concepts. The latter first aims at detecting the web forms already matched in $Adnosco$ which are semantically close to the new web form, and then to perform a fine-grained matching between the new web form and the ontologie(s) which are matched to the closest web forms. To evaluate both methods, we will propose a **benchmark** whose datasets are composed of web forms from related domains (e.g., flight booking, hotel booking, car rental), ontologies, and the materialized concepts between them. Such a benchmark will allow us to confirm experimentally the results presented in this paper.

10. References

Batini C., Lenzerini M., Navathe S. B., "A Comparitive Analysis of Methodologies for Database Schema Integration.", *ACM Computing Surveys*, vol. 18, num. 4, 1986, p. 323-364.

Bellahsene Z., Bonifati A., Rahm E., *Schema Matching and Mapping*, Springer-Verlag, Heidelberg, 2011.

Bramhall P., Hansen M., Rannenberg K., Roessler T., "User-Centric Identity Management: New Trends in Standardization and Regulation", *Security Privacy, IEEE*, vol. 5, num. 4,

2007, p. 84-87.

Cameron K., "The Laws of Identity", http://msdn.microsoft.com/en-us/library/ms996456.aspx, 2005.

Dashlane, "Dashlane", https://www.dashlane.com/download/Dashlane_IFOP_release_2013-03-26_en.pdf, 2013.

Euzenat J., Ferrara A., van Hage W. R., Hollink L., Meilicke C., Nikolov A., Ritze D., Scharffe F., Shvaiko P., Stuckenschmidt H., Sváb-Zamazal O., dos Santos C. T., "Results of the ontology alignment evaluation initiative 2011", Shvaiko P., Euzenat J., Heath T., Quix C., Mao M., Cruz I. F., Eds., *OM*, vol. 814 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2011.

Euzenat J., Shvaiko P., *Ontology matching*, Springer-Verlag, Heidelberg (DE), 2007.

Havard, "Vendor Management System", http://cyber.law.harvard.edu/projectvrm/, 2007.

Higgins, "Higgins, open source identity framework", http://eclipse.org/higgins/, 2007.

Interclue, "Lazarus, Mozilla plugin", http://getlazarus.com/, 2011.

Marc Goodner A. N., "Identity Metasystem Interoperability Version 1.0", http://www.oasis-open.org/committees/download.php/29979/identity-1.0-spec-cd-01.pdf, 2008.

Marie A., Gal A., "Boosting Schema Matchers", *OTM Conferences (1)*, Berlin, Heidelberg, 2008, Springer-Verlag, p. 283–300.

MIT, "openpds", http://openpds.media.mit.edu/, 2013.

Nanda A., "Identity Selector Interoperability Profile V1.0", http://download.microsoft.com/download/1/1/a/11ac6505-e4c0-4e05-987c-6f1d31855cd2/Identity-Selector-Interop-Profile-v1.pdf, 2007.

Talburt J. R., *Entity Resolution and Information Quality*, Elsevier, 2011.

"The UIUC Web Integration Repository", Computer Science Department, University of Illinois at Urbana-Champaign. http://metaquerier.cs.uiuc.edu/repository, 2003.

Wikipedia, "Customer Management System", http://en.wikipedia.org/wiki/Customer_relationship_management, 2013.

Worlfram|Alpha, "Personal Analytics for Facebook", http://www.wolframalpha.com/facebook/, 2014.

Zhang Z., He B., Chang K. C.-C., "Understanding Web query interfaces: best-effort parsing with hidden syntax", *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, SIGMOD '04, New York, NY, USA, 2004, ACM, p. 107–118.