Interaction-based Ontology Modeling for User-Centric Social Networks Environments

Luigi Di Caro
Department of Computer Science
University of Turin
Email: dicaro@di.unito.it

Alice Ruggeri
Center for Cognitive Science
University of Turin
Email: ruggeri@di.unito.it

Guido Boella
Department of Computer Science
University of Turin
Email: boella@di.unito.it

***Abstract*—Social Networks are nowadays the most relevant source of information in terms of scientific challenges and proposed computational models. This is due to the huge availability of user data, ranging from interactions, activities, and multimedia messages. The Big Data era is relatively new, and the emergence of user- and scalibility-centered solutions is particularly influenced by these novel and ever-growing data, that need to be carefully organized to remain manageable. In this contribution, we propose a novel approach to deal with social networks data representation that is able to model such complexity without affecting the flexibility of who can interact within the environment, and how. In particular, we revisit the standard methodology of computational ontologies proposing a framework where objects and agents are defined as compositions of atomic semantic information, avoiding preventive and static identification of the system's players. Our method is inspired by the work of James Gibson, who defined an ecological view of the human perception based on objects' natural *affordances*, in which objects spontaneously give cues about how they can be used depending on the agent who is actually interacting. The idea is that while objects and agents can potentially grow without any constraint, the spectrum of all the individual interactions can be the product of limited (and much more simple to represent) links between users and objects' atomic semantic information. In this sense, if an agent 'x' acts on an object 'y', it means that some property of 'x' are activated by the action (i.e., the user embodies a specific role), and some property of 'y' makes the action physically possible (i.e., it allows the action to be performed). In this paper we demonstrate how an interaction-based ontology view with the use of vector spaces can reduce manual efforts while preserving control of dynamic data in social networks.**

I. Introduction

Social Networks are web-based platforms where users with different interests and properties interact and *live* in a kind of virtual *second life*. Indeed, this has been a real name for one of them, SecondLife, where people used to build *avatars*, i.e., 3D self-personifications in an invented world. The aim of this paper, however, is not to overview all the range of the existing platforms, nor to to make distinctions and, least of all, not to face social aspects and problematics related to the use of such technology. Actually, we want to focus the attention on how these data coming from users interactions in social platforms are of scientific interest in terms of knowledge representation and ontology modeling.

We usually refer to the term ontology as a set of formal descriptions and tools to represent a specific domain (or a part of it) in an objective way. This is usually reflected in a definitions of objects with fixed properties, and relations among them that depict the dynamic aspect of the representation. For instance, we can think at the following description for a generic object A:

> *An object A is defined by some attribute p_1, p_2, and p_3 which can have some values within a specific numeric range like $[1, 10]$ or among a set of nominal values, e.g., $low, medium, high$. Then, A can exhibit the functionalities f_1 and f_2 to the external world, representing its dynamic part, i.e., its behaviour.*

Representing the world by starting from objects and relations between them is a classic and intuitive way to make the things working both conceptually and at an application level. Object-Oriented programming (OOP) is one of the most successful programming paradigms that uses this architecture to represent internal data structures. Each object carries its own functionalities with itself or it inherits them from a superclass. While this perfectly works in several scenarios, we want to stress the fact that the dynamic part of the architecture (let us now use terms like *actions*, *interactions*, *messages*, *functionalities*, and *operators* for identifying this concept) must inhabit inside the objects. In other words, what can happen with an object has to be defined in the object itself. This somehow freezes the high variability of how an object can be used, and, in general, how agents can interact with it. Centering Social Networks data, functionalities and visualization directly on users needs the *rest of the world* to be as much flexible as possible from a representational point of view.

Most of the times, working on objects as main concepts to be defined is both practical and sufficient. However, this strictly depends on the nature of the domain under definition. For example, social networks are extremely dynamic environments where OO-style objects like users, interests, and locations could be secondary with respect to the interactions that make them active and communicating.

Our idea regards an ontological modeling of the behavior of intelligent agents, built on top of the concept of *affordance* introduced by [1] to describe the process underlying the perception. In his work, Gibson [1] claimed that objects assume different meanings depending on the context, and more specifically, taking into account the *animal species* that interact with them. Implementing this concept in a social network environment would lead to ontologies based on *interactions* rather than on objects. For instance, let us quickly consider an example where a user (data creator) publishes on the network

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-369-8

some comment about a hole in a specific street. Then, let us assume we have two types of users (consumers), a cyclist and a public-transport passenger, respectively. While the object under consideration has an high priority for the formers, it can be probably worthless for the latters. In a classic representation scheme, each object-user combination needs to be thought and formalized a priori, manually checking all the possible cases with the relative contraints. Note that this can be extremely consuming in terms of manual effort (creation / management of interventions) and carry to unflexible and redundant representations which do not embed the concept of knowledge sharing. In addition to this, not only objects (like street holes) and user types can create a large space of representation, but other dimensions can be added, further multiplying the problem. For instance, the basis of the network is the type of interaction, i.e., what people can do, creating a three-dimensional space user-interaction-object which results to be untreatable with classic First Order Logic-like representations [2].

Along this contribution, we will talk about ways of thinking at knowledge by means of objective and subjective representations, highlighting limits and workarounds. Afterwards, we will present our idea of interaction-based computational ontologies as an approach to solve some of the discussed issues, proposing an implementation to represent social networks data. We, finally, conclude the paper with a list of future work directions and open problems.

II. Research Questions

Social networks have the need of structuring all the data in efficient ways, not only from a computational perspective, but rather considering conceptual schemes that better enhance the user experience itself.

In real-life scenarios, it is common to find complex cases where data coming from different sources can cross several aspects, ranging from bureaucracy issues to restaurant reviews. Managing both the quantity and the sparsity of the data is the first problem to tackle with advanced techniques. Then, spreading the data to users according to interests, actual and current needs, and with the right priority is even more challenging. Still, not only social networks usually have to notify users autonomously, but they also have to answer to specific user queries. Indeed, the concept of *search* in social networks is crucial and partially different from standard information retrival tasks of common search engines. In fact, the latters have to index data (text, images, videos, etc.) to be retrieved by means of classic few-words user queries, whereas, in social networks, queries connect locations with people, crossing communities, events, and specific time ranges. All this is even made more complicated by the presence of continuously-changing information like hashtags, emotional states, and smartphones application data.

From a computational and ontological perspective, the challenges faced by this contribution are the following:

RQ #1 how to minimize manual efforts in building computational ontologies

RQ #2 how to represent such complex data maximizing the sharing of the whole knowledge in a social network

RQ #3 how to represent the data without affecting the flexibility of objects and agents interactions

RQ #4 how to capture and shape the dynamism and the variability of the interactions depending on who/what is interacting

RQ #5 how to enable smart access strategies in dynamic and multimensional data (fuzzy search, graph search, cross-aspects search, etc.)

In general, in social networks, a multitude of combinations of aspects must be taken into account depending on several factors like time, locations, interests, and so on. These are not well represented by classic paradigms where the world is a matter of objects and relationships, since this does not cope with the explosion of cases to define a priori.

III. Cognitive Background and Related Work

In this section, we overview the main foundations from which our contribution is mostly inspired. Since our proposal has to do with how ontologies can be used in information systems, it is worth to cite important works like [3][4] that deeply describe main issues and state-of-the-art approaches. Fiske and Taylor [5] highlight important features related to Social Cognition, and computational approaches are needed to better fit users' activity.

The starting point of the discussion is the use of formal ontologies. In general, formal ontologies are inspired to the basic principles of the First Order Logic [6], where the world is explained by the existence of defined objects and fixed relationships among them. This belongs to a physical and static view of the world, since this representation is able to treat only the existence of objects and relationships. The same actions are offered to all agents interacting with the object, independently of the properties of these agents.

Our aim is to manage concepts which have different perspectives depending on the kind of agent or species is interacting with them, instead of having an object duplicated in different classes according to the different possible behaviors afforded to different agents. A social-driven ontology would lie between two extremes, as the first-person ontology mentioned by Searle [7].

For example, the door provides two different ways to interact (the set of methods, if we want to use a programming language terminology): a way for a *human* user and on the other side the one for a *cat*. These two ways have some common actions with different implementations, but they can also offer additional actions to their agents or players. For example, a human can also lock a door with the key or shut it, whereas a cat cannot do it. The behavioral consequence of "how to interact with the door" can be "opened by the handle" rather than "pushed leaning on it", and the way the action will be performed is determined by who is the subject of the action.

The second example has a different character, since it refers to a technological artifact, i.e., a printer. As such, the object can have more complex behaviours and above all the behaviours do not depend only on the physical properties of the agents interacting with it but also with other properties, like the role they play and thus the authorizations they have. The printer provides two different roles to interact with it (the set of methods): the role of a *normal user*, and a role of *super user*. The two roles have some common methods (roles are classes) with different implementations, but they also offer other different methods to their agents. For example, normal users can print their documents and the number of printable pages is limited to a maximum determined (the number of pages is counted, and this is a role attribute associated to the agent).

The third example we consider is of a totally different kind. There is no more physical object, since the artifact is an institution, i.e., an object of the socially constructed reality [8]. Let us consider a university, where each person can have different roles like professor, student, guardian, and so forth. Each one of these will be associated to different behaviours and properties: the professors teach courses and give marks, and have an income; the students give exams, have an id number, and so forth. Here the behaviour does not depend anymore on the physical properties but on the social role of the agent.

Mental models have been introduced by Laird [9], as an attempt to symbolic representations of knowledge to make it computable, i.e., executable by computers. This concept is the basis of the most important human-computer cognitive metaphor [10].

Another related work which can be considered as a starting point of our analysis is about the link between the Gestalt theory [11; 12] and the concept of affordance in the original way introduced by Gibson for the perception of objects. Wertheimer, Kohler and Koffka, the founders of the Gestalt movement [12], applied concepts to perception in different modalities. In particular, it is important to remind the principle of complementarity between “figure” and “ground”.

The same concept is applicable in natural language understanding. For instance, let us think at the sentence “The cat opens the door”. In this case, our basic knowledge of what the cat is and how it moves can be our ground to understand the whole figure and to imagine how this action is performed. In other words, the Gestalt theory helps us say that the tacit knowledge about something (in this case, how the cat uses its paws) is shaped on the explicit knowledge of “what the door is”. Following this perspective, the concepts are not analyzed in a dyadic way, but in a triadic manner.

Considering the literature in Object-Oriented programming (OOP), it is worth citing Powerjava [2], i.e., an extension of the Java language where an objective and static view of its components is modified and replaced on the basis of the functional role that objects have inside. The behavior of a particular object is studied in relation to the interaction with a particular user. In fact, when we think at an object, we do it in terms of attributes and methods, referring to the interaction among the objects according to public methods and public attributes. The approach is to consider Powerjava-roles as affordances, that is, instances that assume different identities dependeing on the agents.

Weissensteiner and Winter [13] focus on landmarks contained in texts to analyze their role in the general understanding of routes. Distributional Semantics [14] represents a novel way of estimating kind of affordances at natural language level relying on statistical analysis. Finally, it is important to refer to [15], where the authors demonstrated that natural language grammar and planned actions are related systems.

Dynamic taxonomies [16] exploit a set of instances classified in a taxonomy to create latent connections between nodes belonging to different paths. In fact, if one instance is classified under two concepts on different paths means that there is some link between them that the original taxonomy was not aware of. This approach is useful to browse a taxonomy by iteratively selecting nodes for filtering the data, and in this sense it has some relations with every work on making structured knowledge dynamic and changeable with respect to some context.

IV. The Approach

Social networks are a modern way people use to communicate and share information in general. Facebook, Twitter, Flickr and others represent platforms to exchange personal data like opinions, pictures, thoughts on world-wide facts, and related information. All these communities rely on the concept of *user profile*. A user profile is generally a set of personal information that regard the user in itself as well his activity within the community.

Understanding the reference prototype of a user is central for many operations like information recommendation, user-aware information retrieval, and user modeling-related tasks. In this context, the concept of *affordance* can be used in several scenarios. First, it can be a way to personalize the content to show to the user according to his interests and activity. This is massively done in today’s web portals, where advertising is more and more adapted to the web consumers. Secondly, the whole content shared by ’user friends’ can be filtered according to his profile, in the same way as in the advertising case. Notice that this does not have to do with privacy issues. In fact, a user may be not interested in all facts and activities coming from all his friends. Social networks started taking into consideration these issues, and our proposal regards an ontological modeling of the data that could autonomously and naturally work in this sense.

Commonly, we can think at the interactions in a network as classes managing rules and constraints to match users with fixed categories or objects (the terms *object* and *category* are interchangeable, referring to “things” that “lives” in the network around the users. A scheme of this scenario is illustrated in Figure 1. Notice that this approach creates one class for each combination user-category (when it is semantically allowed), and it produces a large set of unflexible and predetermined interactions to be formally defined.

For example, if we consider the class StreetHole representing the street holes instances in the platform, we need to model all the agents that can interact with it, like CarAgent (a class modeling the instances of people moving with cars), BikeAgent, and so forth. The problem is that, with this methodology, all possible agents and objects have to be defined a priori in the ontology, without an appropriate *uncertainty management*. In our approach, we do not look for a complete coverage of the interacting agents/objects, since we actually do not represent them as physical concepts, while we only manage sets of fine-grained semantic information units that *everything* (i.e., agents rather than objects) can have in a specific context/time scenario.

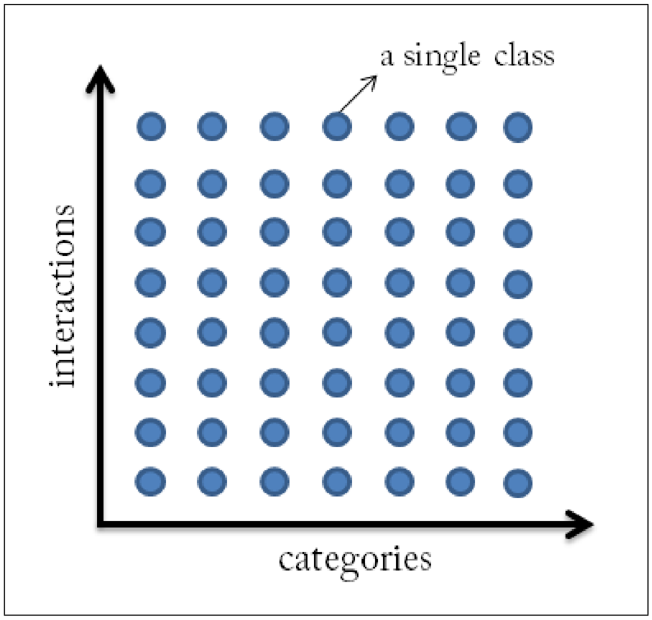

Figure 1. Classic view of single interactions connecting users and categories. Each interaction has its own “life” and it is different from the others, in the sense that it does not share any information nor overlapping degree with the other ones. The distance between the points do not carry any information.

Our idea is illustrated in Figure 2. Objects and users are substituted by the concept of property (i.e., a semantic information unit), on which interactions directly lie. More in detail, each interaction is defined as a set of user features connecting a set of object features, producing area-like representations. This way, the need of constructing classes for managing all the possible $users * actions * objects$ falls into an $m + n$ space, where m is the number of user properties and n is the number of object properties (m and n may have a certain overlapping degree, however).

In this section, we propose a way to model social networks data in a flexible way. As we already anticipated, in most social networks people can participate in the network through a set of interactions. For instance, some of them could be the following:

> $\{$*to buy, to read, to sell, to eat, to drink, to pay attention, to work, to learn, to play, to know, to relax, to participate*$\}$

Each interaction is defined as two sets of *properties* or *features* or *semantic information unit*, for the users and for the objects, respectively. A agent/user or a category/object can be associated to a property with a certain weight. More in detail, the value for a property can be a value in the range $[0, 1]$ representing a degree of affinity within the social network environment, or a nominal value from a given set S (Figure 3).

All the users (also called agents and subjects in our examples and figures) share a set of properties $A = a_1, a_2, ..., a_n$. Some example of user features are:

> *age, sex, marital status, type of work, location*, and a value of affinity for all the objects in the environment (her/his *interests*)

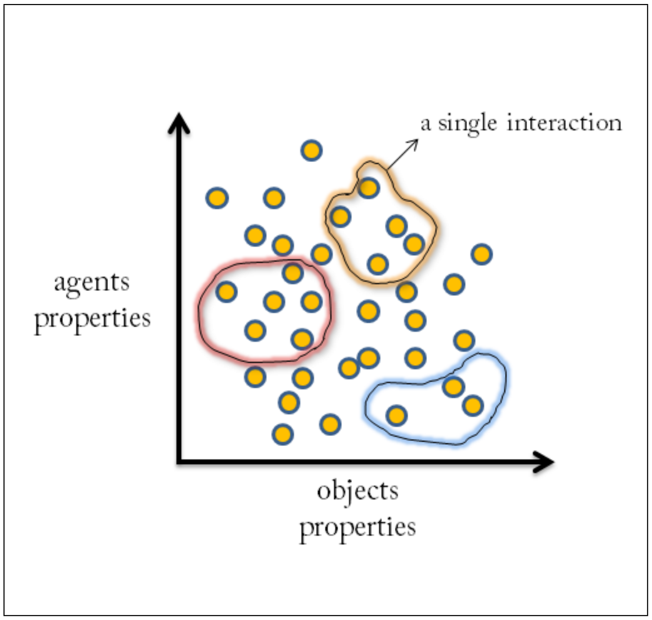

Figure 2. In a property-based interaction scheme, categories and users disappear from the graph since they become simple compositions of features, while the latters constitute the new basis of the representation. Each interaction is thus defined as a set of user/agent properties connecting a set of category/object properties, producing area-like representations. However, notice that each interaction is actually formed as multiple and non-contiguous areas, while the figure has been only created to easily communicate the concept.

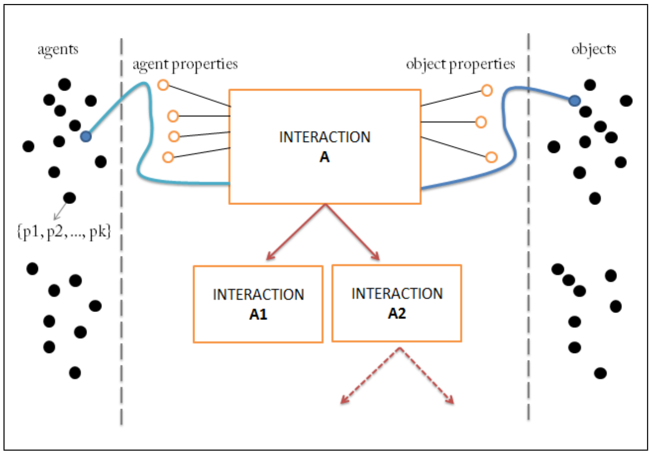

Figure 3. Interaction-centered knowledge representation. Interactions are defined by means of two sets of properties, i.e., for the users and for the categories. Users (black nodes on the left) and categories (black nodes on the right) satisfying these constraints can participate to the interaction. All the interactions are structured in a taxonomy, where subclasses inherit both sets of properties from the superclass. Secondary, users and categoriess are defined as simple sets of properties / features.

In the same way, objects share a set of numeric and nominal property $O = o_1, o_2, ..., o_m$. Examples of them are:

> *bureaucracy, building, city mantenance, sport, education, news, kids, nature, tourism, shopping, lost and found, public transport, personal transport, hotels, restaurants, culture, entertainment, animals.*

An example of object vector is the following, representing a thermal spa in the city centre:

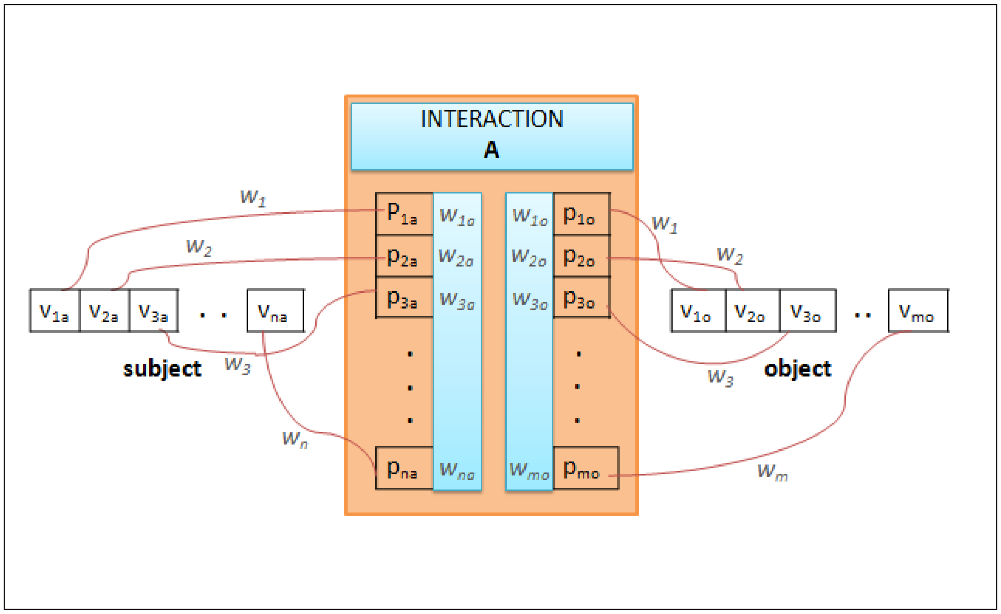

Figure 4. Subjects/users and objects/categories can assume a value in the range $[0, 1]$ for each specific property in the social network environment. All users and objects are thus represented as vectors. In the same way, a specific interaction becomes a two-vectors model that represents the association weights with the subjects and the objects properties, respectively. If a property is not set for a certain user-vector / category-vector / user-interaction-vector / category-interaction-vector, this is treated as a constraint of having a 0-value for any other vector that will be compared with it (otherwise the total similarity value between the two vectors will be set to 0).

object-vector (a thermal spa) *public transport:0.4, bureaucracy:0.0, building:0.2, city:0.5, mantenance:0.0, sport:0.4, education:0.2, news:0.0, kids:0.2, nature:0.7, tourism:0.5, shopping:0.2, lost and found:0.4, personal transport:0.6, hotels:0.3, restaurants:0.6, culture:0.0, entertainment:1.0, animals:0.0*

The wights represent a value of how a specific object is related to a property. In the example, a thermal spa results to be more related to entertainment and transportation rather than to bureaucracy and animals. This way, users and objects are defined as vectors in these two multi-dimensional spaces, according to the Vector Space Model [17]. Notice that, in this manner, objects that present a similar conceptual nature may change its *property status* and so becoming different things depending on the context.

Each interaction, in the same manner, is defined as two vectors of weights (one concerning the user side, and one for the objects), and it can be placed within a taxonomy inheriting all the properties from its parents with some tuning of the weights. Notice that in case of non-numerical attributes, the weights can be numerical transformation obtained by techniques like Multi-Dimensional Scaling [18] and Self-Organizing Maps [19], or by manually-computed ranges.

The first phase concerns the development of the interaction ontology, where the domain experts have to edit a first sketch (even if this can be tuned by users activities dynamically) of the taxonomy of the interactions. Initially, we considered a flat organization where interactions work independently, but the system can work with hierarchy-based constructions as well. In detail, the knowledge engineer has to create the two vectors of the model (the one for the user and the one for the object) for each interaction. An example is shown later in this section.

At this point, once the interaction ontology with all the model vectors are created, a user in the network can act according to the *adherence* between his/her properties with the ones of the existing interactions (their left-side vector in Figure 4), dynamically, and in real-time. The adherence is computed by means of the well-known cosine similarity. From the other side, all the objects are represented as vectors of features as well. One object can be represented by a value of affinity with all the other objects. This is both practical and plausible, since one object can be related to others in some way. For example, the category *public transport* has a significant degree of affinity with the category *private transport*, and it is much higher than what it could be with the category *sport*. There are several ways for computing such graded categorizations in automatic ways also in taxonomy structures as in [20]; however, we think that such process must be done manually (or with a manual support), trying to capture the actual semantics according to the specific domain of application.

To sum up, the initial modeling efforts lie in the configuration of the interactions by weighting user and object vector weights. An example of user-vector model for the interaction *to relax* is the following:

user-vector *age:'any', location:'any', sex:'any', public transport:0.4, bureaucracy:0.0, building:0.2, city:0.5, mantenance:0.0, sport:0.7, education:0.2, news:0.3, kids:0.2, nature:0.8, tourism:0.8, shopping:0.8, lost and found:0.0, personal transport:0.6, hotels:0.4, restaurants:0.6, culture:1.0, entertainment:1.0, animals:0.3*

Then, users dynamically change their feature vector through their own activity in the network (and therefore they constantly change their interaction scenarios). In addition, as in the basic idea of Dynamic Taxonomies [16] by which

ISBN: 978-1-61208-369-8

instances classified under different objects are viewed as latent connections between the latters, a real-time adjustment of the weights is not only done by user-side, but also on the object vectors. In fact, initial manually-constructed object vectors can exploit the real use carried by users activities to find unknown affinity connections (or to moderate the ones already known). This prevents from incorrect configurations in the cold start.

V. Definitions and Validity of the Approach

The entities involved in our proposal are the following:

- **Property**. Also called semantic information unit, it represents the central brick of the world under representation. Every agent/object/interaction is built on top of it.

- **Agent/Object**. It is a set of pairs $< p, v >$ where p is a property and v is a value within its domain $D(p)$.

- **Interaction**. An interaction is a pair of left and right property sets, defining who interacts with what.

- **Interaction Taxonomy**. Interactions are organized in a taxonomical structure such that if an interaction I_p is parent of an interaction I_c, then all left and right property sets of I_p are inherited by I_c.

Agents and objects are compositions of properties, so there is no need to build user- and object ontologies. This *minimizes manual efforts in building computational ontologies* (see Research Question (RQ) #1 in Section II) exploiting the efficacy of the vectorial representations. In the same way, the model *maximizes the sharing of knowledge* since objects and agents use the same feature space (RQ #2). Then, *the flexibility of the interactions is not affected by such representation* (RQ #3) since they directly rely on them by being modeled in the same fashion by two feature vectors. In addition, the actions that the platform can take can be easily defined with constraints on the agent- and object vectors, so it is possible to *shape the behaviour of the social network* (for the same interaction) depending on who/what is interacting (RQ #4). Finally, the use of numerical vectors *completely fits the requirements of smart access strategies*, since it is the model used for queries and retrieval by definition (RQ #5).

VI. Conclusions and Future Works

In this paper, we proposed an idea for representing the knowledge of highly dynamic environments like social networks and Web Sharing sites. Indeed, these kind of information need to be carefully organized to remain manageable while making the interaction itself enhanced. We first started the discussion by thinking at a classic Social Network scenario where users are associated to interests and locations, acting over (virtualized) real-life objects. Then, multiple interactions can take place by means of several combinations of these concepts, thus the knowledge complexity and the relative management becomes interesting as much as it gets harder.

In future works, we will implement these ideas on a real Social Network named First Life, an innovative social network that enables every resident to really live in the neighbourhood by creating a networked platform where citizens can receive and submit useful information. *First Life* has been selected as one of the *Smart Cities and Social Innovation and Communities projects* for the development of smart cities at national (italian) level.

References

[1] J. Gibson, “The concept of affordances,” Perceiving, acting, and knowing, 1977, pp. 67–82.

[2] M. Baldoni, G. Boella, and L. Van Der Torre, “powerjava: ontologically founded roles in object oriented programming languages,” in Proceedings of the 2006 ACM symposium on Applied computing. ACM, 2006, pp. 1414–1418.

[3] N. Guarino, Formal Onthology in Information Systems: Proceedings of the First International Conference (FIOS’98), June 6-8, Trento, Italy. IOS press, 1998, vol. 46.

[4] T. R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing?” International journal of human-computer studies, vol. 43, no. 5, 1995, pp. 907–928.

[5] S. T. Fiske and S. E. Taylor, Social cognition: From brains to culture. Sage, 2013.

[6] R. Smullyan, First-order logic. Dover Publications, 1995.

[7] J. Searle, Mind, language and society: Philosophy in the real world. Cambridge Univ Press, 1998.

[8] J. R. Searle, The construction of social reality. Simon and Schuster, 1995.

[9] P. Johnson-Laird, Mental models: Towards a cognitive science of language, inference, and consciousness. Harvard University Press, 1983, no. 6.

[10] D. Gentner and A. L. Stevens, Mental models. Psychology Press, 2014.

[11] W. Köhler, “Gestalt psychology.” 1929.

[12] M. Wertheimer, W. Köhler, and K. Koffka, “Gestaltpsychologie,” Einfuhrung in die neuere Psychologie. AW Zickfeldt, Osterwieck am Harz, 1927.

[13] E. Weissensteiner and S. Winter, “Landmarks in the communication of route directions,” Geographic information science, 2004, pp. 313–326.

[14] M. Baroni and A. Lenci, “Distributional memory: A general framework for corpus-based semantics,” Computational Linguistics, vol. 36, no. 4, 2010, pp. 673–721.

[15] M. Steedman, “Plans, affordances, and combinatory grammar,” Linguistics and Philosophy, vol. 25, no. 5, 2002, pp. 723–753.

[16] G. Sacco, “Dynamic taxonomies: A model for large information bases,” Knowledge and Data Engineering, IEEE Transactions on, vol. 12, no. 3, 2000, pp. 468–479.

[17] G. Salton, A. Wong, and C. S. Yang, “A vector space model for automatic indexing,” Commun. ACM, vol. 18, no. 11, Nov. 1975, pp. 613–620. [Online]. Available: http://doi.acm.org/10.1145/361219.361220

[18] J. B. Kruskal, “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis,” Psychometrika, vol. 29, no. 1, 1964, pp. 1–27.

[19] T. Kohonen, Self-organizing maps. Springer, 2001, vol. 30.

[20] J. W. Kim and K. Candan, “Cp/cv: concept similarity mining without frequency information from domain describing taxonomies,” in Proceedings of the 15th ACM international conference on Information and knowledge management. ACM, 2006, pp. 483–492.