Published in "International journal on advances in Software", 2018, vol. 11, no. 1-2, pp. 6-17, which should be cited to refer to this work.

The CoWaBoo Protocol and Applications: Towards the Learnable Social Semantic Web

Athanasios Priftis
Information Systems Department
HESSO / HEG-GE,
Geneva, Switzerland
e-mail: athanasios.priftis@hesge.ch

Jean-Philippe Trabichet
Information Systems Department
HESSO / HEG-GE,
Geneva, Switzerland
e-mail: jean-philippe.trabichet@hesge.ch

Abstract— **Web platforms and applications are generating a normalized environment for users to consume information. This process of making the Internet experience "clickable" and "fun" comes at a price: we are less inclined to face important decisions on how applications work, data are handled, or how algorithms decide. This article will examine the possibility of shifting from predetermined results to open and descriptive applications. We inscribe this effort in the context of the Social Semantic Web (s2w) and aim to add a pragmatic approach relying on the importance of humanly created semantics, as a means to fulfil the vision of the s2w. To achieve this we have introduced a descriptive protocol (CoWaBoo) that revisits fundamental web user activities such as search, classification, group formation and valorization of participation. This article will build on the 2017 results from our university group course and, particularly, the prototype applications created through the API of CoWaBoo during the same period. Our aim is to shift our attention from how things end up on the web, to how things *become*, regarding software and applications development. The conclusions of this article will provide further questions for the development of the protocol (CoWaBoo), its applications and the competencies that we need to develop to become actors in the s2w.**

Keywords-social semantic web; learning protocols; learning driven applications; semantic structure; descriptive applications.

I. INTRODUCTION

The growing web and algorithmic reality, where users perform prescribed functions or tasks, is an area that we need to continuously discover and understand. As we will attempt to demonstrate in this paper, the CoWaBoo protocol empowers a human-driven interaction in the web – social space [1] providing a better understanding of the opacity and the impact of these tasks, framed around the term 'digital inequality'. The latter has frequently focused on the distribution of computational resources and skills [2]. How people may be subject to computational classification, privacy invasions, or other surveillance methods in ways that are unequal across the general population, could be in violation of existing regulatory protections [3].

The widely spread culture of opacity of web applications, data or machine learning algorithms is operating a certain normativity: when a computer learns and consequently builds its own representation of a classification decision, it does so without regard for human comprehension. The examples of handwriting recognition and spam filtering illustrate how the workings of machine learning algorithms can escape the full understanding and interpretation by humans, even for those with specialized training [4]. Algorithms, such as those underlying the Google search engine, are often multi-component systems built by teams producing an opacity that programmers who are 'insiders' to the algorithm must contend with as well [5]. The opacity concern arises in the middle of an input - black box - output approach. For the most part, we know how the data are being fed into the algorithm: we produce it ourselves through our activities. We know some of the outputs of the algorithm and can reasonably infer how the algorithm has classified the data. What we do not know is how the 'black box' operates, or which bits of data the algorithm selects and how it uses that data to generate the classifications.

The elements above interact with our understanding of the Social Semantic Web (s2w). S2w can be defined as a space in, which social interactions lead to the creation of explicit and semantically rich knowledge representations. An interaction of collective knowledge systems is able to provide useful information based on human contributions and gets better as more people participate [6]. As web application development is conditioned from the communication structure and technological decisions of the organizations [7] providing them, alternative approaches become crucial.

To sum up, we cannot rely on the modern disciplinary methods and frameworks of knowledge in order to think and interpret the transformative effect which new technology is having on our culture. It is precisely these methods and frameworks that new technology requires us to rethink [8]. Our goal in this paper is to propose an analysis that intersects the current state of opacity and, at the same time contributes to our understanding and application of semantic rules to user created applications.

The following sections of this paper will pursue this analysis. In Section II, we briefly position protocols as an important, but not sufficient, parameter against an opaque, black-box culture of application use and development and introduce the Social Semantic Web mechanisms. Section III is devoted to CoWaBoo, a protocol of building web applications, empowered by specific rules and architecture.

In Section IV, we demonstrate the prototype applications, built by students as proof of work for creating open and descriptive applications, based on the CoWaBoo API. Finally, in Section V, we highlight how such user driven applications validate the rules and functions of the CoWaBoo protocol and how more complex social rules can emerge, while raising further research questions. Our results include presenting the use of CoWaBoo for a specific, knowledge management, application, providing with some early directions on social tagging and user based web ontologies.

II. PROTOCOLS, DATA AND THE SOCIAL SEMANTIC WEB IN A WORLD OF INTERNET APPLICATIONS

The advent of blockchain facilitated the creation of several protocols allowing for the implementation of user driven web applications under the heading of decentralized applications [9]. The most prominent example of such protocols is Ethereum an open-source, public, blockchain-based distributed computing platform, featuring smart contract (applications) functionality [10]. A protocol is an ambivalent space where both opacity and transparency are possible and certain short-term goals are necessary in order to realize one's longer-term goals [11]. Applications running on protocols can use or, sometimes, play down this contradiction but never break away from it. A protocol is never neutral in the sense that all decisions regarding its functions are already set to bring some kind of normality on users (agents) behaviors. In this sense, it contributes to generating a semantic normativity for users to consume information, but not necessarily transforming it. This process is interlinked with the production and reuse of data in semantic mechanisms that are prescribed in accordance with the objectives of their creators.

This use of semantic mechanisms to describe the meaning of data, and enable the description of their metadata, is allowing for more complex user driven analysis. This includes using potentially any data in any format and apply smart data management algorithms as a commonly accessible resource [12]. However, this initial position does not explain how the social happens in the semantic. Having semantically structured data inputs does not guarantee that a social learning is in place or, even, possible. We will turn to the Social Semantic Web (s2w) to understand how this could possibly happen. S2w aims to add a pragmatic approach relying on description languages for semantic browsing using heuristic classification and ontologies, emphasizing the importance of humanly created loose semantics as means to fulfil the vision of the semantic web. Manuel Zacklad, Jean-Pierre Cahier and Peter Morville define s2w as the pace layering of ontologies, taxonomies, and folksonomies to learn and adapt as well as teach and remember [13].

Various investigations have been published regarding the evolution of the social and semantic web, including the development of ontologies for tagging [14], the extraction of ontologies from social network graphs and folksonomies [15], collaborative ontology evolution and reasoning over tags [16]. The social and semantic web is appearing as a mixture of multi-disciplinary information that is evolving within an open environment [17], making it difficult to initiate a learning process. In these efforts, there are various possible social approaches for solving the problems of user driven ontology evolution for the semantic web. Users could create folksonomies or flat taxonomies to document their information. Social Network Analysis (SNA) allows the ontology to be extracted from the tags and be reused into topic maps or ontology stores. These organized tags could be manually analyzed to create a more sound ontology. Another approach is to create a system for self-governance where the users themselves create the ontology over time in an organic fashion. All of these approaches could start out with an empty ontology or be seeded manually or with an existing ontology.

The protocol that exemplifies user-oriented ontologies is SSWAP (Simple Semantic Web Architecture and Protocol). SSWAP is an architecture and protocol for semantic web services. It uses the W3C standard of OWL (Web Ontology Language) to ground a web services model on computable semantics and logic by formulating an architecture and a protocol. The HTTP API allows developers to generate SSWAP RDF/XML graphs using JSON [18]. The main application, based on SSWAP, is called iPlant. The iPlant Collaborative seeks to enable data-driven scientific integration both within the enterprise and across web resources, including widely used programs of general interest and niche programs for specific needs [19]. IPlant's Semantic Web Platform is developed as the technological conduit for integration across various plant resources. It is by having software layers handle data, service syntax and semantics that iPlant seeks to free the scientist to focus on data and service use.

Application development comes at the junction of these diverse and interdisciplinary efforts. With the multiplication of user driven applications, we could tap into a large part of the motives and rules of the created interactions. When it comes to information and internet technologies' mediated systems, the technical arrangement of the underlying interaction platforms do have an impact on the social cognition processes [20]. The process of channeling more or less sophisticated representations, from opinions and likes/dislikes, to tags, sentences, or documents are some of the elements that need to be, continuously, examined.

However, a single instance of data (i.e., a tag) can never be semantic on its own. Semanticity is captured at the logic of organizing data within a system of relations or rules applied to this system. Several problems encountered in ranking, documentation and automated search can be handled by semantic conceptualization and graph theory [21], while many others persist. Our position is that instead of relying entirely on automated semantics with formal ontology processing and inferencing, humans could (learn to) build their applications. While the semantic web enables integration of business processing with precise automatic logic inference computing across domains, a Social Semantic Web could be used for a more socially oriented technology interface, allowing for a more divers interaction between various objects, actions and their users. The CoWaBoo protocol, including this paper, aim to contribute to this exact area, in between semantically oriented protocols such as SSWAP, or transaction oriented ones like Ethereum.

As importantly, we need to connect s2w with a clear educational demand; the people who craft the code, that determines all the million material ways in which we interact, are able to consciously articulate the things they believe, agree or disagree [22]. The Social Semantic Web presents us with a possibility to understand and affect the design of software and applications and, eventually, how could it be reformed. The CoWaBoo protocol and its applications are set to intermediate this effort.

III. The CoWaBoo protocol and its open API

CoWaBoo starts as a concept, leading to an ambition of understanding and affecting user actions in a socio technological context. The CoWaBoo protocol considers the addressee, user of these wider computational systems made up of processes, technologies or networks, as a possible actor in their design. As the interaction needs to be open to further development, the code of the CoWaBoo protocol needs to apply the following default rules [1]:

- Assure the lucidity of past (as stored data), present (as current data collection, or processed archival data), and future (as both the ethical addressee of the system and potential provider of data and usage).
- Store objects with a semantic description that generate functions for applications.
- Demonstrate that the code's mediation can be rethought, researched for intervention, contestation, as a tool for the un-building of other code/software systems.

The overarching concept of the protocol is to formalize an always-editable space realized by the rules set above. However, this space stays vague if we do not test and experiment its utility. This is why we bring in the collective observatories application, as one of the many possible applications, built upon the protocol. In this application, the protocol would store and allow us to recover a general index of a given subspace (observatory) and then be able to navigate through the different versions of the entries in this observatory. To achieve this within the application, we introduce two more layers of representing information reusing data from the protocol: a) the use cases, or the way we propose users to explore the possibilities of the protocol, through the creation of collective observatories and b) the graphic representation of classified information. The use cases need to be concrete with a measurable result. Users can search information that communities have already curated and form groups. The graphic interface attempts to address how users will experience the above.

In terms of communication, data and account handling the CoWaBoo protocol adopts the following approach. All data are stored on the InterPlanetary File System (IPFS), a P2P (peer to peer) storage protocol, with its current state available in the application. Account creation is based on Stellar, a blockchain based, open source protocol, used for all transactions between users. CoWaBoo is utilizing a NodeJs server with a Stellar Javascript SDK to provide the CoWaBoo API with a way to communicate directly with Stellar. Every time that a new member is subscribed, a new Stellar account is created. This account receives the minimum amount of lumens (Stellar currency) to work properly as a CoWaBoo account. Once the account is created, it automatically give its consent to carry out currency exchanges created by the "bank", or main wallet, of CoWaBoo in Stellar. Consequently, all transactions (i.e., votes in groups) are stored on the Stellar blockchain infrastructure creating an exchange community with a cryptographic Public Address and Secret Key. While Web 2.0 applications tend to prescribe our participation, in CoWaBoo we seek to re-open the discussion on the group rules and their results. Therefore, all new entries or definitions are openly editable, as long as her/his entry is voted in a group. In the following paragraphs, we will discuss in detail the utility of this function and demonstrate if and how the default rules are applied.

Once an entry is added, modified or deleted, the observatory generates a new version of itself with a unique, cryptographically generated and traceable identity. This system of versioning, creating new instances for every edition on an entry, follows the blockchain paradigm. Blockchain as a distributed, cryptography boosted, database technology is a thing of the 80s, which computational capacity of our time brought to full implementation with the Bitcoin deployment. Blockchain can be understood as an implementation of distributed ledgers that comes with a unique set of possibilities in its design. It opens up the way to shared databases, where multiple entities can transact, with no or some trust between them, co-existing with no intermediation.

The CoWaBoo protocol reproduces the main blockchain synchronous properties, as described above in the following generic approach: a) accessible and affordable shared data with resilience through replication and no single point of failure and control, b) where multiple entries are possible, c) based on the possibility of disintermediation. Blockchain implementation comes with more interesting feature: d) application based transactions, or smart contracts [23]. Going back to our collective observatories application, we will try to point out the exact process and code that demonstrate the above. The CoWaBoo objects described by the protocol are organized in the following categories:

- Dictionaries or Observatories
- Entries
- Propositions
- Users
- List of all users
- List of all tags

The process of communication and connection between these objects in the protocol is depicted in Figure 1.

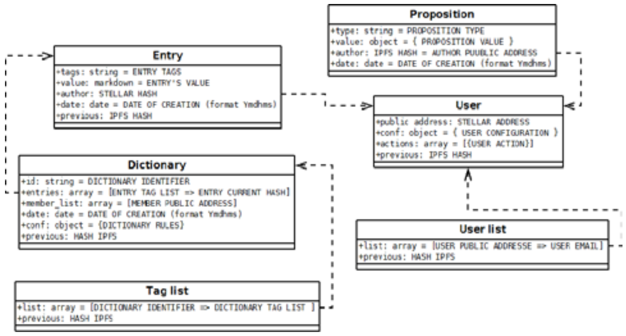

Figure 1. The connections between the CoWaBoo protocol objects

This loose semantic architecture of the protocol, as well as, the collective observatories application proposed above have one important result. CoWaBoo can be understood as a re-documentation effort opening to possible, iterative transformations of user drive applications through its API and the way it organizes the user access to its objects. Figure 2 offers an overview of the CoWaboo API and its structure: tags, observatories, entries, users and accounts creation. It consists of several functions and elements, also presented in the following chapter: The user-oriented view of the CoWaBoo API is demonstrated Figure 2.

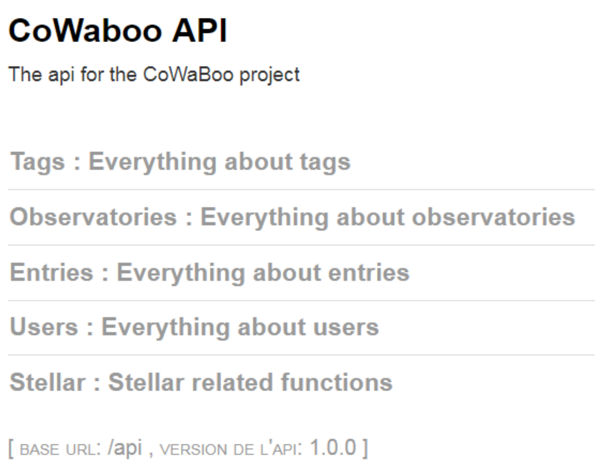

Figure 2. A user-oriented view of the CoWaBoo protocol and its API

In the following section, we will demonstrate how user driven applications are intermediated by the CoWaBoo API. This activity provides us with a key resource for understanding semanticity, as the logic of organizing data within a system of social relations. These relations will be deployed with user proposed rules functioning on code. Beforehand, we will present the use cases that define the protocol and present the context for the CoWaBoo API utilization. In other words, how it prescribes the construction of user driven applications with socio semantic decisions and contributions.

IV. Implementing the CoWaBoo protocol through use cases and user applications

A protocol is never neutral in the sense that all decisions regarding its functions are designed to bring some kind of standardization on its user's actions. Thus, it becomes crucial to describe, in more detail, the use cases of the collective observatories application and illustrate how the interaction with the protocol will take place. We can think of the use cases as a descriptive middleware, positioned between the protocol and the collective observatories application, appearing itself to its users, as its initial implementation.

Build your story: A prompt to click on the start button and move to an empty text area is the first step. The empty text area is destined to be the user's notebook, potentially filled up with search results discovered using the connected APIs for search on tag, bookmarks, articles and existing entries, themselves, linked to the visualize tags and stories in our observatories use case. Keywords for initial search lead to tags looking to visualize existing stories in the CoWaBoo observatories. The user selected data are inserted in the text area, as part of the actual user story. This tentative story-result can stay local, private and unfinished, which is posted in an observatory, or become the first entry in a new possible observatory. Adding a post in an existing observatory is subject to a verification process depending on the rules of the observatory: self – validated means that the post is validated by the user, peer – validated means validated by a "vote" from someone in its existing members.

Edit a story: A click on the full text search button of the application connects the user's keywords, then transformed to user tags, used to visualize existing stories in observatories or observatories themselves. These tags are linked to the visualize tags and stories in observatories use case. The CoWaBoo user can select them as content for further editing into the text area.

Start a community of transactions (group): A click on the community button initiates a community creation function with the possibility to add emails, each participation verified through the related email account and attribution of both public and secret keys. The group creation launches the possibility of starting: transparent, intra-group transactions with all group users being informed on energy limits to credit or stored value of the group. Group participants use the secret key to perform transactions, while the results of the transactions, as well as, the user balance remain publicly linked to each user's public key. New members propose themselves through direct demands to join an observatory (group) or through accepted stories: when observatory entries are accepted they become a member, while, his/her public key is added to the group users for further transactions.

Editing tags: CoWaBoo treats tags as distributed objects, recasting the tag object as an autonomous transaction providing its user with an opportunity to redefine, rebuild and redistribute through the work of others. Tags in CoWaBoo acquire a multiple meaning as they simultaneously represent:

a) “Tags to be”: user typed search keywords in the text area leading to tags used by other users.

b) Keywords that are then selected as tags, leading to educated choices of entries description, or proposed entries or definitions for observatories.

c) Tags are treated as semantic elements pointing to entries in observatories (list of tags per observatory).

d) Names of observatories are treated as tags (list of observatories). They are proposed to users for consultation before creating a new observatory.

Propose or “vote” an entry or a member (validation & valorization). Each story is accepted as an entry (definition) when posted and voted on by at least one group member as a verification (the fastest reply - user is considered for attributing the value of the transaction). Each accepted entry is tokenized with one (1) energy (limit +10 for every user), while a vote for an entry uses 1 energy, (- 10 for every user). The variation in personal energy is, initially, anonymous but transparent, connected to each user’s public key.

Visualize tags and stories in observatories: This use case is connected to user keywords (full text search) becoming tags and visualizing an index view with linked tags, entries and observatory names.

The following are some important questions that will guide the presentation of the initial results of the protocol and its collective observatories application: Firstly, do tags serve at the same time as descriptive keywords, linked data (to stories) and ongoing collections (observatories names)? How do they connect entries and observatories? Do they provide some kind of navigation through the information initiated by the application and stored by the protocol? Secondly, the protocol does not promise, or highlight, a completed story or observatory but a possibility of creating stories and editing all products in future events. How is this appearing in the existing digital space? Can the CoWaBoo protocol and its linked applications serve as an experimental understanding at semantic, representational and process levels in a context of collaboration and knowledge production?

Before attempting to answer these questions, we will present how the CoWaBoo API prescribes the construction of user driven applications. As mentioned before, the CoWaBoo API allows users to experiment and develop applications on the CoWaBoo protocol. This process seeks to create a space where a user centered semantic construction becomes possible, under the defining and implementation of **tags, observatories, entries, users, blockchain enabled accounts and transactions**.

Tags: The user can call (GET) all the tags and access the documentation in the provided URLs. Tags, as semantic objects, have multiple functions: form an id for entries, act as descriptive keywords or provide a list of data available. Figure 3 demonstrates how the use of a simple API function (GET) allows the user to call all tags with their semantic description, see exactly how this function works (CURL) for the CoWaBoo protocol and access the generated URL available to reuse.

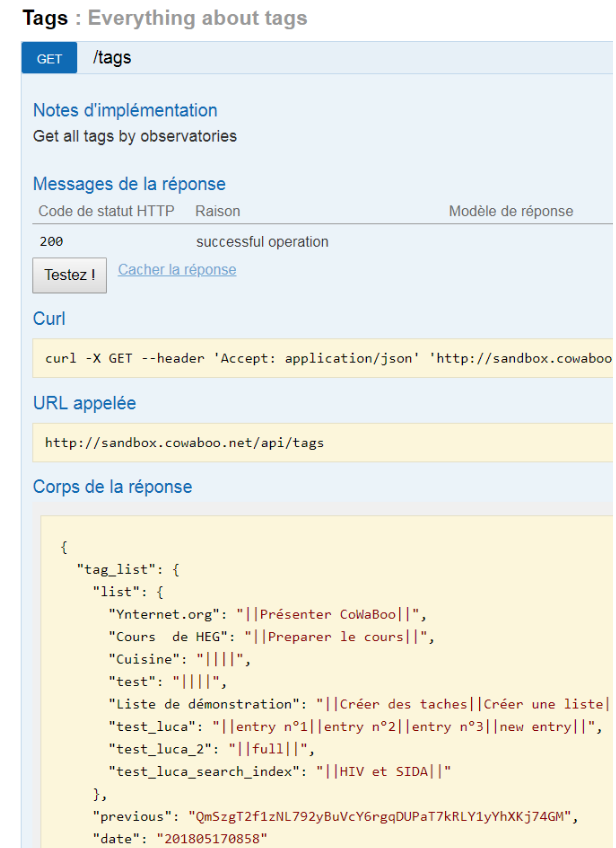

Figure 3. The CoWaBoo API call for all tags

Observatories: This API section acts as the main logical container of whatever information or concept is to be expanded by the forthcoming entries of the application. In a sense, this is an initial hierarchization and organisation of the application’s data as shown in Figure 4.

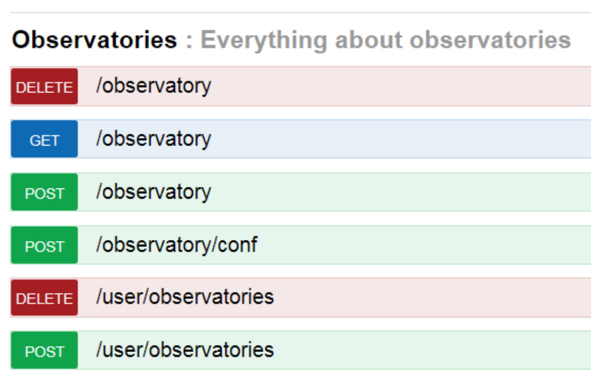

Figure 4. The CoWaBoo API observatories

Entries: The Entries section is the core semantic repository of the protocol. This is where JSON, XML or any other structure form of user-oriented and application rules data can be saved, recuperated, edited, deleted, reposted and, infinitely, versioned through its cryptographically created editions. Figure 5 summarizes the CoWaBoo API entries.

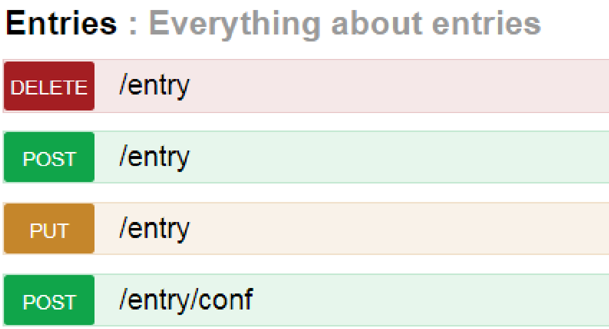

Figure 5. The CoWaBoo API entries

The Entries section (Figure 6) come with a detailed API description: this is where the user Secret Key (author) is used to add and validate data posts and add tags as entries i/d or keywords.

Figure 6. The CoWaBoo API entries' section structure

Users: Users, as shown in Figure 8, are handled by creating a new, public and blockchain (Stellar) linked, account. This creates a public and a secret key communicated automatically to every new user. Both are then reused to validate observatories, entries and perform transactions in applications (i.e., votes).

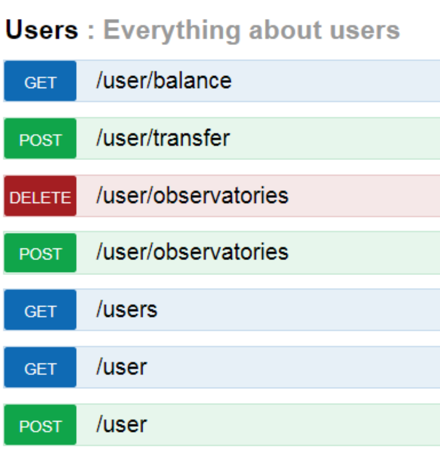

Figure 7. The CoWaBoo API users' section

Figure 8 demonstrates how the user's transactions are taking pace using their public and secret Key. The amount can be any value the user wants to transfer through the Stellar blockchain protocol. It corresponds to a generic currency called "energy" created by the CoWaBoo protocol and allowing users to define it as credit, votes or other. In summary, "Energy" is a subdivision of Stellar's Lumens cryptocurrency and is implemented as a rule centrally administrated by the protocol maintainers.

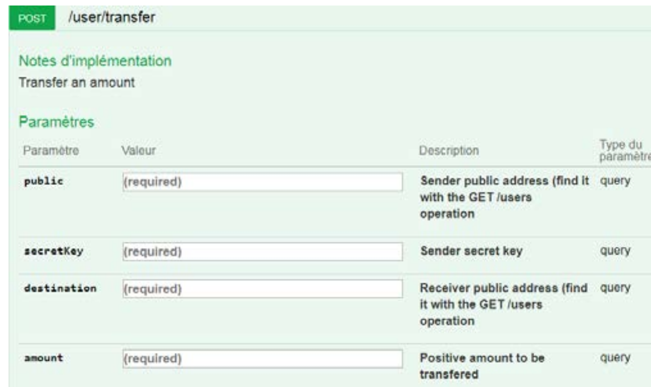

Figure 8. The CoWaBoo API and its users' transactions section

Stellar use: As mentioned above, Stellar related functions are adding the possibility to access user related transactions. This function, described in Figure 9, gives the possibility to users to create their own currency for their applications use and test all their transactions with the public Stellar blockchain.

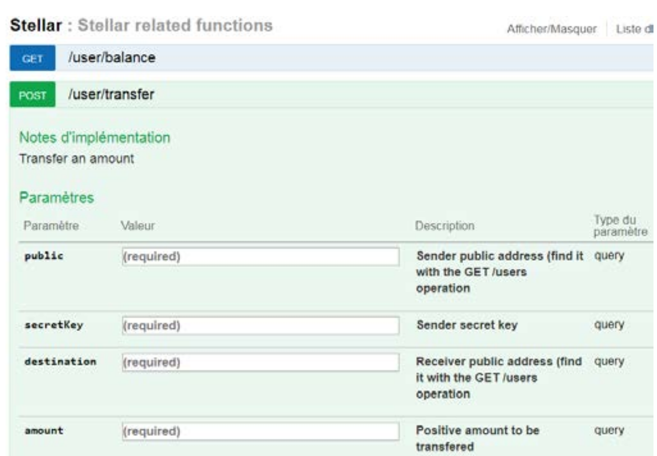

Figure 9. The CoWaBoo API and its Stellar related functions

Different instances of the API are easily created to provide a new space for users to develop their applications. At this point, every new CoWaBoo API instance is controlled centrally. However, this is not a bottleneck for future creation of applications because there are no data or other information kept locally and the creation of API instances could be handled directly by its users. This architecture, although rudimentary in its current state, provides a novel experience to users on creating new applications. Users are free from the data based infrastructure and account generation process to focus on the semantic representation of their application's data and processes.

V. Early results and remarks on how to evolve the social-semantic mechanisms in app building

Our results are organised and presented in the following areas: firstly, we present with some documentation around the learning and development context: this is a testing effort structured around groups, for a minimum prototype application building activity. Secondly, we will highlight an example of an application built in the CoWaBoo protocol context, demonstrating the social semantic potential of this process. Finally, we will try to summarize and organise our contributions to the development of a user driven s2w understanding.

The design and implementation of the CoWaBoo protocol started in 2015, as a research effort on building a bottom up, social bookmarking process. The design activities and results of our 2015 - 2016 experiments including focus groups with experts and students testing various versions of the application, led to the creation of the presented collaborative observatories application on the CoWaBoo protocol. This application was introduced in our university courses during the spring semester of 2016 for further testing and analysis. The use of the CoWaBoo protocol and API, as a user-driven, application development space was introduced in 2017. This new experimentation phase was organized with twenty-four students of the University of Applied Sciences in Geneva during a dedicated course on Digital Business Technologies. This group of, 3rd year, bachelor students were invited to work in groups of three, around a specific thematic area (i.e., Blockchain, Bitcoin, Wikipedia, Airbnb, Uber, Free and Open source software, Open licensing, Open innovation) using the CoWaBoo collective observatories application as the main space of documentation and the CoWaBoo protocol and API for their own prototype applications' development. This activity included two concrete areas for further evaluation. Firstly, participants were asked to organize and present an introductory course in class. The course is documented in the collaborative application based on CoWaBoo. Thus, it was important to demonstrate how the use of an application relates with the CoWaBoo protocol. Secondly, each group had to design and prototype an application based on CoWaBoo. While reviewing below these results, we will try to highlight several social semantic elements related to web applications' building.

In the following paragraphs, we are presenting selected results regarding the use of tags from students in their respective observatories. This is linked to the use of an application related with the CoWaBoo protocol and is connected to our Editing tags use case, described in Section IV. Our understanding of tags as descriptive keywords, linked data and ongoing collections is the basis of the following remarks.

The first remark targets the way the information of each observatory, eight in total, is presented, accordingly to a certain tags selection. Let us use the "blockchain" observatory to see how its tags are visualized:

Blockchain (theme and name of the CoWaBoo observatory) with selected tags:

|| DAO || France || IDE || Parlement || analysis || badge || banques || bdd || bitcoin || blockchain || chain || concept || crypto-money || ethereum || finance || finance on blockchain || fonctionnement || governance || infographie || peer-to-peer || plate-forme || politique || presentation || questionnaire || reference || smart contract || technologie || wikinomie || wikipedia ||

This static and hierarchized representation of an observatory with its related tags is a straightforward use of all social bookmarking applications on the web and gives access to related user entries.

The second remark is on how each tag is linked to other tags. This function uses data, coming from all other CoWaBoo observatories, connecting them as linked tags with the word blockchain. In this sense, we are advancing to a networked relation and navigation of various observatories through commonly used tags.

Linked Tags (in English and French) to other observatories for the "tag" blockchain include:

Cowaboo, properties, avantage, bitcoin**,** monnaie**,** concept**,** fonctionnement**,** infographie**,** smart contract**,** reference**,** presentation, wikinomie**,** badge**,** questionnaire**,** Wikipedia**,** peer-to-peer**,** innovation ouverte**,** plateformes**,** copyleft**,** creativecommons**,** smartcontract

A third remark has to done on the entries (stories) that are being created and edited in the respective observatories, with the tag blockchain in each entry. Entries using the tag blockchain include:

- CoWaBoo

|| CoWaBoo || blockchain || properties ||

- Bitcoin

|| avantage || bitcoin || blockchain || monnaie ||

- Blockchain

	blockchain		concept		fonctionnement		infographie	
	blockchain		smart contract	
	blockchain		reference	
	blockchain		presentation		wikinomie	
	badge		blockchain		questionnaire	
	blockchain		smart contract		wikipedia	
	bdd		blockchain		peer-to-peer	

- Innovation Ouverte

|| R&D || blockchain || innovation ouverte || plateformes ||

- Copyleft

|| blockchain ||copyleft || creativecommons || smartcontract ||

In this representation, we observe that multiple tags act as the title of an entry. The word blockchain is now used as a search keyword that leads, subsequently, to entries that include it in its tags.

A fourth remark needs to be placed around the use of the word blockchain in all the observatories that CoWaBoo is hosting. **Observatories using the tag CoWaBoo include:**

- CoWaBoo
- Bitcoin
- Blockchain
- Innovation Ouverte
- Copyleft

This presentation highlights the observatories that can interest a user with a search interest in the blockchain: before clicking on a specific entry, the CoWaBoo user has a hierarchized overview of the areas that could be connected to his/her research.

These four remarks derive from the daily, dynamic use of the CoWaBoo protocol and its social bookmarking – collective observatories application. They summarize the possibility of an application, like our collective observatories, to use the CoWaBoo protocol's functions in order to reclassify user data. The protocol therefore formalizes all the necessary functions:

- Data storage, recording new data in an observatory: entry, member list, configuration
- Ensuring the current state of an observatory
- Possibility to add, change and delete data from an observatory
- Registration and acceptance of new members
- Vote counted and executed through blockchain intra-community transactions
- Configurability and re-applicability of the rules of the group in the protocol itself.

It is important, to describe if and how the default rules are applied. Once an entry is added, modified or deleted, the observatory creates and stores a new version of itself. In Figure 10, we depict the current state of the blockchain observatory. This includes its i/d, entries, members, date, configuration (public or private) author and, most importantly, the path (hash) to its previously stored version. This process applies to all data stored from the CoWaBoo protocol and generated from the collaborative observatories application: observatories, entries and member list.

Figure 10. A CoWaBoo (blockchain) observatory stored on IPFS

As already mentioned the InterPlanetary File System (IPFS) is a content-addressable, peer-to-peer hypermedia distribution protocol. Nodes in the IPFS network form a distributed file system. IPFS plays a crucial role in CoWaBoo acting as a public ledger of all posted or edited data. The same goes for any rule, or post. All changes in membership, authorship and rules of this observatory are re-traceable in the blockchain logic of the protocol with the previous version always available. Content wise, things are significantly different, compared to a standard social bookmarking application. Entries are taking more of wiki form with a descriptive text and links. Figure 11 shows the entry linked to the following tags:

|| France || Parlement || banques ||finance || politique || technologie || entry

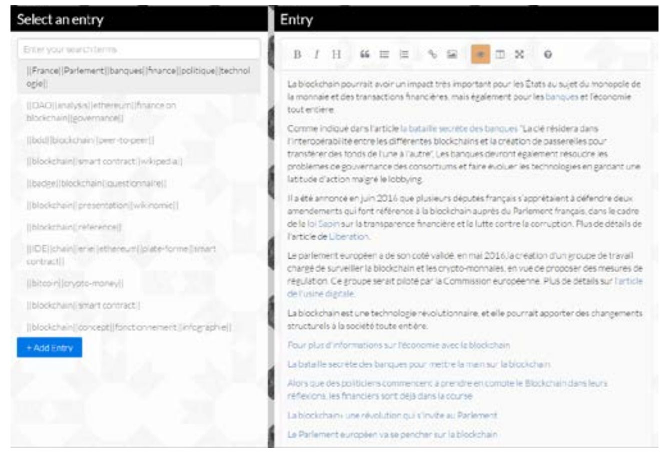

Figure 11. A CoWaBoo, collective observatories application, entry

There is a clear shift, in all observatories, to adding text and sources in entries and forming them more as stories. This is due to the discussion related to CoWaBoo lecture on social bookmarking and the evaluation criteria but it goes hand in hand with our initial anti-click-culture approach. In the table below, we summarize how CoWaBoo introduces the redocumentation process in steps.

TABLE I. THE COWABOO REDOCUMENTATION PROCESS

Step title	Description	Process
Search and Post	Users are invited to search in existing communities, discern knowledgeable others, edit text and post result in existing groups	Search through tags on existing community curated resources Select resources and edit a story Publish on a group a shared "block of information" with tags ⇒ Create unique link (URI) that could be posted on selected web services
Reuse and Curate	Initiate a process of critical scrutiny and group content production for participants	Engage in a group - observatory, where an exchange between users, data and group rules is possible. Provide feedback on group's entries Contribute to the implementation of the group's collective observatory Provide history of group decisions and documentation

The second area of results announced in the beginning of this section is around the social semantic web elements in application building possibilities of the CoWaBoo protocol.

The user applications created in 2017 complete the results presented above. The overall estimated effort to create each application is one week, as a total effort for all group members. The main idea behind this initiative was to introduce them to an early application building and confront them with the design and implementation opportunities and decisions that come therewith.

To illustrate our analysis we will use the ShareMyPlace app as a representative example of the eleven applications created and available in the CoWaBoo applications repository. The landing page of this application is presented in Figure 12 and acts as a list of private parking spaces available in the city. The use of this application can see the entries involved (time, locality, contact) and reserve its place.

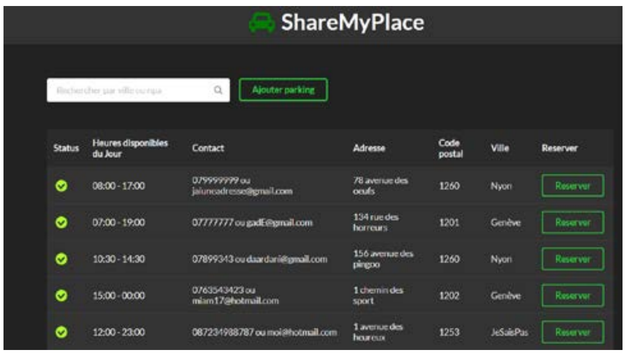

Figure 12. The ShareMyPlace CoWaBoo application: landing page (list observatories)

The first observation that we can do while clicking on the Reserve button is that we can "pay" in credits. These credits are blockchain transactions linked to the CoWaBoo protocol possibility to use its currency as credits or votes. Once the function triggered, the user needs to identify as a blockchain user and complete the transaction. Figure 13, shows the interface of the ShareMyPlace CoWaBoo application.

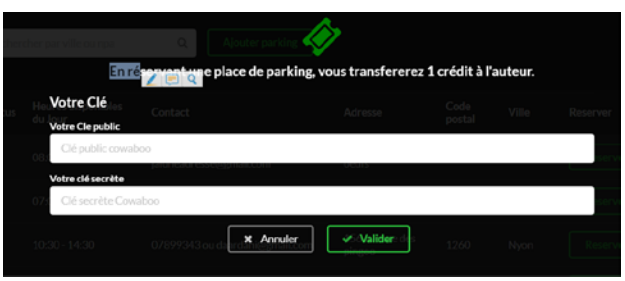

Figure 13. The ShareMyPlace CoWaBoo application: transactions' interface

The second observation is about the use of the protocol to organize the post of entries in the application. As shown in Figure 14, entries can have multiple values and connect to observatories. Their posting to the application involves a Secret Key validation, meaning that users need to identify, describe and propose a new parking space as a tentative or permanent entry (transaction) to an observatory.

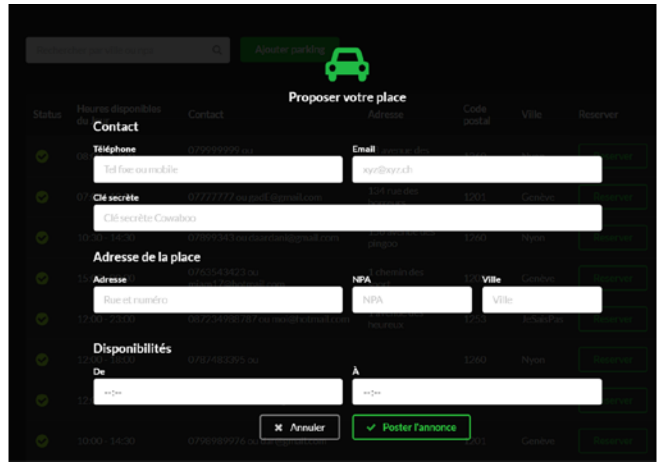

Figure 14. The ShareMyPlace CoWaBoo application: entries

The main social semantic lesson to retain from this prototype application development is the way its observatories (dictionaries for the protocol) are organized. The application designers opted for the creation of multiple observatories: one observatory was used for each parking space proposed in the application. Thus, the developers preferred to make a distinctive and separate entry for each value leading to a linear, heavy and non-efficient creation and data storage process, as shown in Figure 15.

dictionary:{id: "parking1ede38f6-446c-002e-1cc9-2493858aba0e", entries: {,…},…}
dictionary:{id: "parkingcdd6d6f1-5987-911b-ea45-9cdff84a68b1", entries: {,…},…}
dictionary:{id: "parking0906c3b5-6f17-0c10-58d9-11ebf252e77d", entries: {,…},…}

Figure 15. The ShareMyPlace CoWaBoo application: multiple observatories, one observatory used for each parking space

Multiple entries can be inserted in multiple observatories, as shown in Figure 16. This implies a minimum of effort to organize and describe the application data and rules in a coherent and structured way. This would mean, for example that one observatory could accommodate all values of an entry through a semantically structured text.

dictionary:{id: "parking0906c3b5-6f17-0c10-58d9-11ebf252e77d", entries: {,…},…}
author:xxxx-xxxxxxxxx@etu.hesge.ch
conf:[]
date:"201705231318"
entries:{,…}
id:"parking0906c3b5-6f17-0c10-58d9-11ebf252e77d"
member_list:["xxx.xxx-dasilva@xxx.hesge.ch"]
previous:"QmPhqxXC6xkLWYwVvpnjUboyEtGaSxngyStdjUpsPff9Gg"

Figure 16. The ShareMyPlace CoWaBoo application: multiple entries in multiple observatories

We will profit from this lesson learned in order to provide an analysis of a different experience of application building through the CoWaBoo protocol. This example falls under the social semantic web elements in application building possibilities of the CoWaBoo protocol but comes from a personal student effort during its Bachelor degree work during the summer of 2017. This new application is called GraphtoLearn and it is being designed as a learning path based on the indexing and organisation for several keywords. It starts out as a thesaurus of learning courses in the Information Technology area, augmented with the following possibilities:

- custom and dynamic term search based on user participation (user and entries reputation and institution rules)
- user participation and entries annotation comes with assigned reputation rules
- 3D visualization of the results search that attempts to demonstrate the above rules

This indexing - learning application demonstrates the rise of social semantic structures empowered by the CoWaBoo protocol. We do not intend to analyse all the GraphToLearn possibilities but focus on the organisation of entries and reputation, rules as a part of its social semantic web elements.

Firstly, let us have a look on Figure 17 and, particularly, on how the architecture of GraphToLearn is organised. The API CoWaBoo is central in all data storage and reputation rules (votes on user actions and entries). GraphToLearn comes with the ambition that its semantic entries can describe and incorporate existing structured data; new user generated data and automatically harvested data (i.e., Google, Wikipedia). The selected data flows can be used to enrich the GraphToLearn entries, as well as, boost the application's reputation and search algorithms.

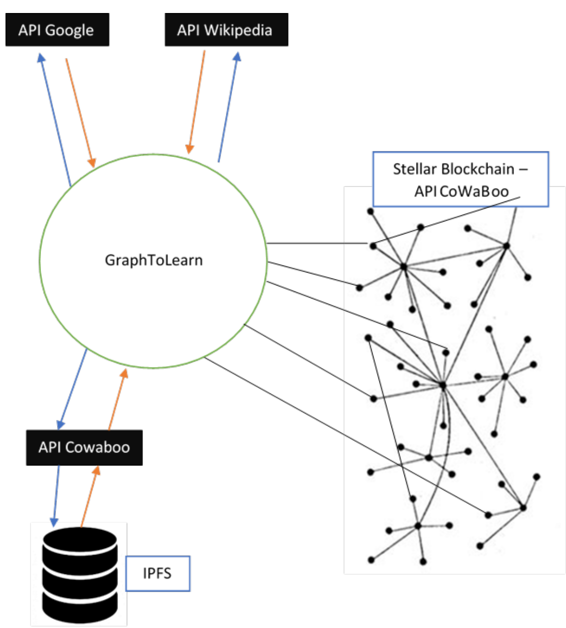

Figure 17. The GraphToLearn initial architecture

In this architecture, GraphToLearn is presented as a concept where the aforementioned possibilities are inscribed. This is a crucial step to our analysis, a clear intention to explore the social semantic in knowledge organisation and production. It shows that the application developer is inspired by the CoWaBoo protocol to deploy a more complex vision of user-oriented taxonomies, including a system for their self-governance. The organization of its entries, as presented in Figure 18, take a semantic turn with the following choices:

- One observatory including all entries
- Each entry includes all the needed information in a JSON format

```
{
  "dictionary": {
    "id": "Words",
    "entries": {
"QmST3o7XqmJLG8ZHa1M81vFH1ZHgAocammPEjBsR9s
zaho": {
        "tags": "||ActiveX||", -> nom du terme
        "value":
"{\"name\":\"ActiveX\",\"type\":\"Terme\",\"source\":[\"CRI\"
],\"definition\":\"\",\"explications\":\"Contrôles utilisés en
programmation web pour permettre d'animer des
pages.\",\"context\":\"Technique\"}",
        "author": "xxx@etu.hesge.ch", -> la personne qui l'a
inséré
        "date": "201707101559", -> date de création du terme
        "previous": null,
        "conf": []
      }
    }
```

Figure 18. The GraphToLearn: observatory and entries

Each entry is structured and described as presented in Figure 19. Various elements (data with their metadata) of the application are described and posted in the "value" area, following the design of the CoWaBoo API.

```
"QmShYR6hQrD83pqh7ndofhi8ipQ7QC7w6t6N95dMn
13Soe": {
"tags": "||MIB||",
        "value":
        "{\"name\":\"MIB\",\"type\":\"Acronyme\",\"so
        urce\":\"IDEC Metafichier\",\"modules\":\"471,
        498\",\"definition\":\"Management Information
        Base.\",\"explications\":\"Arborescence de
        variables stockant les états des matériels et
        logiciels selon le protocole
        SNMP.\",\"context\":\"Technique\"}",
"author": "xxx@etu.hesge.ch",
"date": "201707111415",
"previous": null,
"conf": [] -> qu'est-ce que je pourrais mettre dans conf ?
}
```

Figure 19. The GraphToLearn entry structure

Figure 20 demonstrates, through a JSON representation and structure, how the value of the entry is organised and posted in a specific entry (see Figure 20).


```
"{
  "name":"MIB",
  "type":"Acronyme",
  "source":"IDEC Metafichier",
  "modules":"471, 498",
  "definition":"Management Information Base.",
  "explications":"Arborescence de variables stockant les états
   des matériels et logiciels selon le protocole SNMP.",
  "context":"Technique"
  "commentary":"",
      "review":"",
         }"
```


Figure 20. The GraphToLearn JSON representation (data in a single entry)

There is a second area, where the socio-semantic intention of the developer appears in GraphToLearn: the reputation rules for both users and entries. We will present, briefly, the reputation rules, not as a complete technical approach, but as a token of the reflection that the CoWaBoo protocol has allowed for.

Setting up the GraphToLearn reputation makes an algorithmic search possible, allowing the application to sort out the proposed terms and highlight terms that have a better reputation in its search result. This element improves search by distinguishing terms that are related to the term the user wanted to search for. It is reducing search time by looking at the reputation of each term that would be best suited to the term initially sought.

To implement this system of reputation GraphtoLearn defines certain criteria in order to be able to evaluate each term (Term based reputation). It sets a corresponding weight for each criteria. The weight allocation is based on the sub-division of the CoWaBoo protocol currency. The main principles of the user reputation include modifications, new contributions, validating, or not, the work of others. A user who proposes an accepted entry gains one (1) energy if the entry matches the expectations of one (1) or more collaborators and is taken from the tokens of those who voted to accept the entry. Another possibility is that a user who proposed on entry loses one (1) energy if the entry does not match the expectations of one (1) or more collaborators and is distributed to those who have to report the content. Following this process, energy (the CoWaBoo protocol currency) is distributed over the latter. This set of rules is an integral part of the application and can be revisited by its users.

These early reputation rules presented above, affecting reputation of terms and users, are not to be considered as compete reputation systems. They are a clear indication that the CoWaBoo protocol offers its users an accessible possibility to experiment and deploy such rules a in a concrete application. By lowering the barrier of users accessing such features we aim to make the social semantic more user oriented and trigger considerations about the competencies that application developers need to develop.

VI. Conclusions and Further Work

S2w does not imply a high level of “automation of the meaning” with formal ontologies processed by automated inferences, but focuses in situations where a semantic need is translated with the technology. Human beings need to stay in the loop, interacting during the whole lifecycle of applications, for both cognitive and cooperative reasons [13]. S2w is taking a new turn with the rise of P2P (peer to peer) storage protocols and the open public blockchain realm. Users need to come up with new analysis and development tool and face in a more learnable way web platform, protocols, applications. The semantic normativity in platforms and applications around us is not designed, with its users. However, it is influenced by the data the users themselves produce, while using them.

This paper tried to add more details and use cases on how the CoWaBoo protocol aims to reverse our habits to consume prescribed information and describe how user-oriented processes could take place [1]. We used the previous experience of social semantic web efforts to understand how this can happen and brought user driven applications as concrete examples to study this. We focused on the importance of humanly created semantics in web application development, as a means to bridge the gap between opaque algorithmic platforms and user designed, semantically rich, data representations. The GraphToLearn reputation system, appearing in the previous section, is a clear indication that complex socio-technical mechanisms do not need to be opaque or incomprehensible.

During our activities with our university students, we were surprised to notice their leapfrogging possibility to confront with this novel application building. Expressing ideas, documenting their evolution and coding them becomes possible with limited effort. The non-efficient creation process of these applications, equally, surprised us: a lot needs to be done to transform users into s2w actors. In addition, it became clear to us that the social semantic web elements in application building are part of the possibilities provided by the CoWaBoo protocol. Elements such as custom and dynamic term search based on user participation and entries reputation, implementation of group participation rules, individual or group entries’ annotation are part of these possibilities. Several problems encountered in ranking, documentation and automated search can be handled by a more detailed semantic conceptualization of these applications. This could lead us to an era where the s2w includes not only taxonomies and ontologies but real life, bottom up applications, using powerful user designed validation systems.

Although we can give not a binary answer as to whether the CoWaBoo protocol and applications provide us with a semantic and representational tool, specifically, in a context of collaboration and knowledge production, we have demonstrated how such a process could be initiated. Further work on the use of the protocol and its applications is scheduled. This work includes the multiplication of similar use cases, during the spring semester 2019, within a targeted course in the Information Systems Department (HEG) of the University of Applied Sciences in Geneva. This involves reusing the protocol through its use cases and functions, experimenting on:

- More complex group rules and valorization of transactions between participants in various groups. This should include testing of the default rules of the protocol and evaluation of the transactions functions in the application, while leading to group rules and results editable and possible to change from everyone, as long as her/his entry (definition) is voted in a group through the CoWaBoo currency.
- An alternative search experience based on GraphtoLearn and a more educated understanding of community resources as a reference to information search. Our goal is to stimulate competences as penetrating intelligence, keen perception and sound judgment in community driven curation.
- Creating new applications, scenarios and early implementations, based on the CoWaBoo protocol API (including both IPFS and Stellar protocols). These scenarios can be deployed using the protocol and its rules, or being inspired by it.

Finally, we believe that understanding web applications as potential open and descriptive protocols is a crucial step towards more transparency, less opacity, in our digital era. We intend to continue our research both as a way to unmask current opacity in digital technologies and experiment on new tools that could support collaborative and critical competencies.

References

[1] A. Priftis and J-P.Trabichet, “CoWaBoo: a descriptive protocol of learning driven applications,” The Ninth International Conference on Mobile, Hybrid, and On-Line Learning (eLmL 2017), IARIA, March 2017, pp 94-100 ISSN: 2308-4367, ISBN: 978-1-61208-541-8

[2] E. Hargittai and A. Hinnant, “Digital inequality,” Communication Research, vol. 35, no.5, p.602, 2008

[3] S. Barocas, “Data mining and the discourse on discrimination,” The Data Ethics Workshop, Conference on Knowledge Discovery and Data Mining (KDD 2014), KDD, Aug 2014. [Online]. Available from: URL: https://dataethics.github.io/proceedings/DataMiningandtheDiscourseOnDiscrimination.pdf

[4] E. Fisher and M. Naumer, “Information grounds: Theoretical basis and empirical findings on information flow in social settings,” in New directions in human information behavior, A. Spink & C. Cole, Eds. Amsterdam: Kluwer, pp. 93-111, 2006.

[5] J. Burrell, “How the machine ‘thinks’: Understanding opacity in machine learning algorithms,” SAGE Publications, pp. 23-30, 2016.

[6] K. Weller, “Knowledge Representation in the Social Semantic Web,” Knowledge Information, Berlin : De Gruyter, vol. 3, pp. 82-93, 2010.

[7] S. Cicero - Why Platforms need to be Engines of Learning — Stories of Platform Design. [Online]. Available from: https://stories.platformdesigntoolkit.com/platforms-are-engines-of-learning-4f7b70249177#.ygl8vmabw44A

[8] F. Coffield, D. Moseley, E. Hall, and K. Ecclestone, “Should we be using learning styles? What research has to say to practice,” London: Learning and Skills Research Centre, 2004.

[9] V. Lehdonvirta, & A. Robleh, A.. “Governance and Regulation.” in M. Walport (ed.), Distributed Ledger Technology: Beyond Blockchain. London: UK Government Office for Science, pp. 40-45, 2016.

[10] V. Buterin, V. Applications. Ethereum White Paper (2013). GitHub. [Online]. Available from: https://github.com/ethereum/wiki/wiki/White-Paper#applications

[11] A. Galloway, Protocol: How Control Exists After Decentralization, MIT Press, 2004.

[12] Y. Marketakis, N. Minadakis, H. Kondylakis, K. Konsolaki, G. Samaritakis, M. Theodoridou, G. Flouris, M. Doerr, “X3ML Mapping Framework for Information Integration in Cultural Heritage and Beyond,” Int. Journal on Digital Libraries, pp 1-19, 2016, dOI 10.1007/s00799-016-0179-1

[13] J-P. Cahier, M. Zacklad, A. Bénel, H. Zaher, C. Lejeune, et al. “Hy-pertopic : une métasémiotique et un protocole pour le Web socio-sémantique“. 18eme journées fran-cophones d’ingénierie des connaissances IC2007, July 2007, pp.217-228, 2007.

[14] T. Gruber, T. Gruber, "Ontology of folksonomy: A mashup of apples and oranges," Intern. Journal on Semantic Web and Information Systems, vol. 3, no. 1, pp. 1-11, 2007.

[15] C. Cattuto, D. Benz, A. Hotho, G. Stumme, J. Baumeister, M. Atzmüller, "Semantic analysis of tag similarity measures in collaborative tagging systems," LWA volume 448 of Technical Report, Department of Computer Science, University of Würzburg, pp. 18-26, 2008.

[16] J. Liu, D. M. Gruen, J. M. Bradshaw, H. Lieberman, S. Staab, "Between ontology and folksonomy: a study of collaborative and implicit ontology evolution," in IUI, ACM, pp. 361-364, 2008.

[17] R. Baeza-Yates and B. Ribeiro-Neto, “Modern Information Retrieval,” Boston, MA:Addison-Wesley Longman Publishing. Co, Inc., 1999.

[18] The SSWAP HTTP API documentation. [Online]. Available from URL: http://sswap.info/api/Home.html

[19] S. A. Goff et al., “The iPlant Collaborative: Cyberinfrastructure for Plant Biology”, Frontiers in Plant Science 2, pp. 1-16, 2011, doi: 10.3389/fpls.2011.00034

[20] C. Roth, “Socio-Semantic Frameworks. Advances in Complex Sysems,” in World Scientific, pp.135, 2013.

[21] P. Morville, Ambient Findability: What We Find Changes Who We Become, O'Reilly Media, Inc. 2005.

[22] S.D. Roy and W. Zeng, Social Multimedia Signals, Springer, Cham, 2015.

[23] N. Szabo, N. “Formalizing and Securing Relationships on Public Networks”. First Monday, vol. 2, no. 9, 1997, doi: http://dx.doi.org/10.5210/fm.v2i9.548