Uniscript: A Model for Persistent and Incremental Knowledge Storage

Adorjan Kiss, Joël Quinqueton

Uniscript: a Model for Persistent and Incremental Knowledge Storage

Adorjan Kiss
LIRMM CNRS
161 rue Ada
34392 Montpellier, France
kiss@lirmm.fr

Joël Quinqueton
LIRMM CNRS
161 rue Ada
34392 Montpellier, France
jq@lirmm.fr

ABSTRACT

We present in this paper a model of personal knowledge representation for lifetime storage.

In the model we separate the knowledge layer from the resource layer. The knowledge layer consists of a network of atomic knowledge units situated in space and time. Resources are data packages (bit sequences) that can be rendered by some device into any human-perceivable form. The two parts complement each other: the knowledge network can be seen as annotations of the resource base (multimedia store) while resources can serve as means for the interpretation of knowledge units as well as a way to index and access them.

For the knowledge network we propose a simple formalism that we consider could support the emergence of a language capable of describing increasingly complex situations of the real world and, by time, to represent any information that is expressible by natural language.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems—*Human information processing*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Relation systems, Semantic networks*

General Terms

Design, Theory, Human factors

Keywords

Personal knowledge modeling, Lifetime storage

1. INTRODUCTION

Traditionally, Knowledge Representation (KR) models are intrinsically connected to reasoning. Represented information was supposed to serve precise purposes, or to allow construction (deduction) of new information with automatic methods [9].

Boosted by the ubiquity of personal computing, new researches orient to decoupling knowledge from expected usage. Among these, there are projects that defend the idea of lifetime storage [7]. However, this orientation raises specific problems. Maintaining the consistence of stored knowledge, access to it, and above all correct interpretation of the stored content must be projected to an undefined future.

In this work we propose a model to represent knowledge for lifetime storage, founded on incremental construction using atomic units of "subjective reality" called *stances*. With this model we try to demonstrate the possibility to describe reality through spatio-temporally limited units, and to separate knowledge from data designed to render them interpretable.

A possibility of deployment is shown with a prototype implementation on PDA.

CARPE'04, October 15, 2004, New York, USA.

2. STORING PERSONAL KNOWLEDGE

We experience lately an ever increasing demand for new and simplified methods to manage and centralize personal data, pushed by the expansion of a multitude of digital devices which can communicate or access the network more and more easily. New organizing possibilities are being proposed by projects as Placeless Documents [5], Haystack [8], Lifestreams [6], trying to free the users from the unnatural constraints of classical file and hierarchy-oriented methods, focusing on semi-structured data in the form of annotations upon the raw information stored in documents.

With our approach we try to go a step further and challenge the necessity to use abstract collections such as documents for personal information. Information as "My sister is allergic to antibiotics" or "Julie moved to Paris last week" could be important to be stored in a personal knowledge base, without being easily attachable to a collection. In fact, we start from the question of what knowledge is, what makes personal knowledge representation different with respect to generic knowledge representation.

2.1 Confidence of knowledge

Words like knowledge, belief, intuition, opinion describe states of a human mind that persist for relatively long time and that guide his actions during his life. To any of these states the human can easily associate an empiric measure: *confidence*, or *certainty*. This measure is in correlation with the probability foreseen by the person that future events might occur that could disturb these states.

We suggest that during his development since his early childhood a human mind develops structures corresponding to aspects or pieces of reality he considers unique. Some of these structures may eventually become stable, resisting in time to new events that the person perceives, while others which turn out not to be "fit" are eliminated [4]. New pieces of knowledge that are created can then be attached to structures that proved to be stable enough. Sometimes, "greater" structures - that were used to connect with many new structures - must be revised due to new events: a process that can be perceived as *surprise*. The larger is the structure to be destroyed, the bigger the surprise. (Note that this is not the only way to quantify surprises. Other measures related to emotional impact, or life priorities of the subject may count as well.)

As he advances with age, the size of surprises a person experiences decrease continually. Until his adult age and even afterwards, a human mind gathers a certain number of structures that become very stable, and upon which a large number of other structures depend [10]. Certain authors would call these structures *commonsense knowledge*, an association we prefer to avoid - we will explain shortly why.

2.2 Validity of represented knowledge

Representing knowledge comes down to writing it in a language or communicative medium. In order to allow this, knowledge must be *decomposed* into identifiable pieces. Once represented, a piece of knowledge loses most of its connections with other knowledge that are doubtlessly connected with it in the human mind. Among them, there are answers to questions such as: which were the events that led to its formation, how was it deduced, which would be the eventual events that could make the person question the pertinence of that knowledge?

In knowledge representation in computer science, any representation method had to face problems due to decomposition of knowledge into pieces: the evolution of the world led to accumulation of inconsistencies within the represented knowledge. These problems were collectively referred to as knowledge maintenance or maintenance of validity.

It is clear that in our context of lifetime storage, it is necessary to reconsider the issue of knowledge maintenance. As suggested by the section before, it is important to identify knowledge of highest certainty.

2.3 Eternal knowledge

A simple question comes up from the above considerations: Which is eternal knowledge?

Biased by the history of knowledge representation, and more profoundly by major philosophical currents, the most obvious answer to this question coming to one's spirit is that eternal knowledge consists of perfect classifications, rules and regularities that govern our physical world. That is, non-situated knowledge.

Our alternative answer is that knowledge can be constructed from representation of phenomena that are situated (limited) in space and time. As noted by Jean Rostand: "Les théories passent, les grenouilles restent." (The theories pass, the frogs remain.)

A major problem is raised by representing situated knowledge: situated phenomena are just too numerous. We can't hope to represent everything. And since we have dropped the necessity to make explicit the purpose of representation, it is essential to have a guiding line to help select what to represent. It is given by the combination of 3 criteria: *importance*, *certainty*, and *uniqueness* - they are detailed next, after presenting the model.

3. REPRESENTING SITUATED KNOWLEDGE

We propose a model to build a network from situated knowledge pieces. The atomic knowledge unit is called *stance*. Connections between stances are oriented links with precise semantics.

The reference point to the model is the person who wishes to capture some of his knowledge. He is presented as an observer who contemplates the world, and whose attention is caught by some events and phenomena.

3.1 Stances

A stance is a personal view of the observer of an entity or phenomenon from his reality, which is situated (has a limited extension) in space and time. It is supposed to be traceable in some way during a time period. From the observer's point of view, it can be seen as a cutout of the space-time continuum surrounding him, which he chooses to regard as a whole, in its uniqueness.

Stance is the central notion in our model, which we needed to well distinguish from other notions that already bear heavy connotations, like *object* (too restrictive), *concept* or *entity* (that do not reflect situatedness). The word stance (originated from pp. of Latin stare) suggests two aspects: First, a phenomenon must have a set of stable features in order to be individualized. Secondly, that the observer feels he has a representation of it which is stable, and he does not expect any surprise to challenge its pertinence.

Typical stances could be:

- Individual objects. Humans naturally delimit real-world objects. We can also individualize and remember some of the objects that we feel to have an importance, associating them a period of existence in the world.

- Parts of objects. Some features of objects we can find to persist during a period of time, possibly during the whole existence of the object. A part that can be individualized can be considered a stance.

- States of objects. Most often during the existence of an object we can separate a number of states of it. When such a state presents a particular interest, it can be stancified.

- Events. If we accept that events are interactions between two or more objects, the reunion of states or parts of the objects that take part in the interaction may be considered a stance representing the event. During the event, the interacting objects may keep or lose their identity. Note that this definition of event is somewhat counter intuitive and not exactly conform to some of the best accepted ones, such as Allan's definition which states that an event must involve at least one object over some stretch of time or involve at least one change of state [1]. Let us just assume that if an event involves only one interesting object, (or other participants are not noteworthy) we will consider it a state, since it must have non-null duration in time - at least long enough to allow recognizing it.

- Groups of stances. A number of stances may share regularities and may be naturally viewed as a whole. If the observer accepts the group as finite, he can refer to it as a stance. Of course, the members of the group can be distant in space or in time. That is, the 4D cutout of space-time is not necessarily a contiguous interval in space or in time.

3.2 Stance examples

Imagining the 3D space compressed into only one dimension, an object could be seen as a 2D shape in space-time as in the diagram: t_0 is the moment at which the observer considers the object came into being, t_1 is when it ceased to exist. The form of the shape suggests that it may have changed place, shape, size or any of its features during its lifetime. Example 1, *The apple tree near my window.*

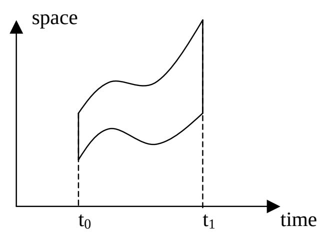

Figure 1: A primitive stance seen as a cutout of space-time

A particular case is a part or feature of a stance, which the observer can distinguish throughout the whole lifetime of the including stance. Example 2, *The root of my apple tree.*

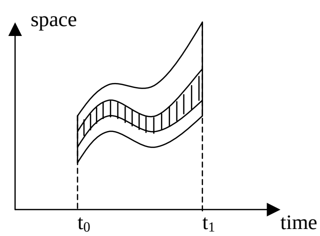

Figure 2: Part or feature of a stance distinguishable throughout its whole lifetime

Another particular case is a state, or period of the lifetime of a stance that can be distinguished. Its space extent may be as large as the one of the including stance. Example 3, *The apple tree in blossom during last spring.*

3.3 Finiteness of stances

Situatedness implies that a stance must occupy a limited amount of space (that may vary in time) and that it must have a finite lifetime (period of existence). At the moment of representation, stances can be of two kinds: existing stances and ended or past stances. A third type, that can be imagined, future stances (that did not yet begin) cannot be represented as such: they do not satisfy the criteria

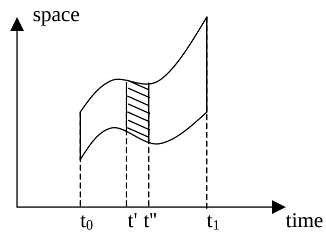

Figure 3: Delimitable state of a stance

of certainty - one can never be completely sure about what will happen in the future. We will come back to this issue in future works, at buffer storage.

In time, stances are bordered by a beginning and an end. Beginning is the moment at which it assumed its identity, from the point of view of the observer. For example, a person watching an island may believe that the beginning in time of the island was the moment when it emerged from sea.

End of a stance is a transformation into other stance(s) identifiable by different criteria, or into something non delimitable. As a stance is unique, the end is irreversible. Once a stance transformed, it will never exist once again: the end is a change of identity. In case of a reversible transformation, it is also natural to see the entire sequence of changes as states of a more generic stance, existing as long as the changes keep happening.

Borders of stances are subjective: an observer is not constrained to agree with anyone else about the position he places the borders. For instance a person may consider the existence of the apple tree since its seed dropped from the parent three, while another observer prefers the moment at which the seed started to sprout. The same applies to its extent in space: the moss growing on the tree may be seen as part of it by some and distinct by others.

Moreover, the observer must not be precise with his delimitations - the borders can be rough or fuzzy: an observer may identify as a stance branch of the tree, without being able to sharply delimit it from the trunk of the tree.

3.4 Individuals, groups and classes

Collections of stances can be perceived as unique whether they are numerable or not - it is enough for the observer to be able to delimit it, and to be confident of the persistence of its unifying criteria. For some groups, it may not be obvious whether its identity comes before (is more important than) the identities of its members, or the other way around. For instance, seeing two apple trees that grew from the same root, one may hesitate to see it as one or two individuals.

Certain groups can be extended to individuals yet inexistent or unknown to the observer. In this case we can talk about classes: a *class* is the set of all instances an observer has found to belong to the group during his life. The number of these instances is necessarily finite, thus the class is limited in space and time.

With this agreement on classes, contrarily to some groups, it is impossible for a class to exist before identifying any of its instances. Thus, descriptions of things that exist only in an abstract form only in the mind of a person cannot be stancified before any instances of them existed. This is the case for instance with artifacts, which are conceived for a specific purpose. They existed as plan in the mind of their creator, hence the temptation to consider artifact classes to precede their instances.

Exactly the same way as shifting between individual and group view of stances, one can shift from group to class and back. An extinct species of animals for example, could be easily reduced to a group if found that only a few specimens existed that developed in isolated conditions.

The point we wish to make is that in our context, having the world contemplated through the viewpoint of an observer, no sharp difference can be drawn between *individuals*, *groups* and *classes*. This is why we believe it would be unnatural to make a distinction between these concepts in the core of our model.

4. STORING KNOWLEDGE WITH UNISCRIPT

Stances of an observer can be stored on a physical support. Storing comes down to associating a unique location (number or address) to each of the stances. Allocating numbers is a definitive operation - it cannot be undone. Next, persistent relations between stances can be identified, and materialized in the support by oriented links.

As a remark here, the very sense of "knowledge" lies within the connections. Without connections, we could not talk about representing knowledge, but about storing data, at most.

Coming back to "commonsense knowledge", we are convinced that they consist of a form of stances and links, which became deeply anchored within the mind, and which a person carries along without much change till the end of his life. And this doesn't resume to abstract and procedural knowledge as the term may suggest, but to individual pieces of reality with their relationships.

Stancification in Uniscript - identifying and writing down stances - is guided by the following principles:

- Principle of implicit importance. Each stance should be found important by the observer, and the interest of a stance should be expected to last. No stance should be meaningless - not correspond to a natural delimitation of the world.

- Principle of certainty. The measure of confidence in the existence in reality of each stance should be as high as possible. The observer should be confident that a surprise that would contradict the existence of the stance has practically no chances to happen.

- Principle of uniqueness. Before recording a stance, one must make sure it is not already represented. For this, he can use the resource layer to access the stance network and search through the neighborhood of existing stances that might be connected. This principle makes the strongest difference from other knowledge representation methods, including natural languages, hence the choice of the Uniscript name. On the possibility to fulfill this principle depends the pertinence of the model.

All these principles are appreciations of the observer. Therefore they cannot be fulfilled to hundred percent. A user's success to master his stance base is directly dependent on the correctness of his appreciations.

4.1 Connections

Recorded stances can be connected through oriented links. Two kinds of links are used to reflect the physical structure of the world: *containment* and *transformation*. A third link type, *revision*, can be used to deal with with surprises - that can happen, even if one has stuck as much as he could to the guiding principles.

Containment links are used to mark spatio-temporal inclusion. A stance S_1 is said to be contained into another stance S_2 if the spatial extent of S_1 was inside the limits of S_2 during the whole existence of S_1. This is the only criteria for containment links. It can be used for apparently different relationships (as part-whole, membership or instance-class relations) - providing that complete inclusion in space and time is respected. In the previous examples,"tree in blossom" and "root of the tree" stances are contained in the "apple tree" stance.

Transformation links reveal temporal precedence between the connected stances. They are used to mark changes in identity of stances. If there is a transformation from stance S_1 to S_2, S_1 is said to be the predecessor and S_2 the successor. For instance, "tree in blossom last year" may be the successor of "tree in blossom two years ago". The extents of two stances linked by transformation are completely disjoint;

Since theoretically both of these link types could be drawn between many different stances, it is up to the author to identify the most significant ones. In fact, similarly to stances, links are persistent, so confidence, importance and uniqueness principles apply to links as well.

Finally, changes in one's view of delimitations caused by surprises can be captured using revision links. Nevertheless, revisions are reserved for exceptional use only. For example, if one had 2 different stances for two people, Dr Jekyll and Mr. Hyde, each having its own history, and later found that he needs to revise them.

4.2 Formal properties

The 3 link types form 3 subgraphs of the main memory. Among formal properties of these link types owing to the properties of physical reality, it may be noted that both relations are transitive and antisymmetric. Antisymmetry of the transformation is vacuously true, due to the non reversibility of the world, because it is impossible to have a stance transform back into its predecessor. Similarly no circuit can occur in the aggregation graph, except between stances with the same space-time extent. More precisely, two stances are *equivalent* if they have the same space-time extent. This case is theoretically possible, but would translate as redundancy in the knowledge base. In addition, containment relation is reflexive, thus partial order.

Another noteworthy property comes from the fact that contained stances cannot "outlive" their container. Considering that S is the current set of stances, let $G_a = (S, A)$ be the graph of the containment relation, and $G_t = (S, T)$ of the transformation relation. If $(x, y) \in A$, $(y, z) \in T$, then $(x, z) \notin A$. It can be interpreted that when a stance transforms (loses its identity), it is supposed that all of its members disappear or transform as well. This means that membership relation in which the member may keep its identity after the container has disappeared, cannot be directly represented with a containment link. For example "the wheel of my car" is not contained in "my car" if the wheel may keep its identity after the end of the car. A solution in such a case is to consider a period of the life of the wheel which is contained in the car.

4.3 Connection examples

A situation like: “Today I went swimming after work”, could be described as shown in the figure 4. In the figure, numbers in parentheses are unique IDs of stances. Dotted arrows represent composition links and solid arrows represent transformations. Stances with a label outside the parentheses have text resources associated (as presented in the next section) that should help restoring their meaning.

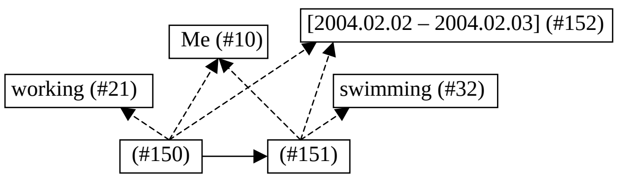

Figure 4: Representing a simple situation

The situation shown here is only meant to illustrate the semantics of the links. Its representation may include conventions that a person may not feel comfortable to make at the beginning of his knowledge base. For example, stance #32 represents here all the states of beings that the observer ever got to know about, that he recognized as “swimming”. Even more special is the case with “today” - how could it be seen as delimitation in space and time. For this, one could consider a geographical place, say Paris, being a stance: existing since the moment it was founded. Then, he could think of “today in Paris”, as the state of the Paris during a period which most of its inhabitants agreed that was this day. And then, he could say that “today” is the grouping of similarly defined states of all the places he ever knew about.

A second example shows how knowledge can be augmented without affecting the existing links. Adding that “I was driving from work to swimming” can look like in figure 5:

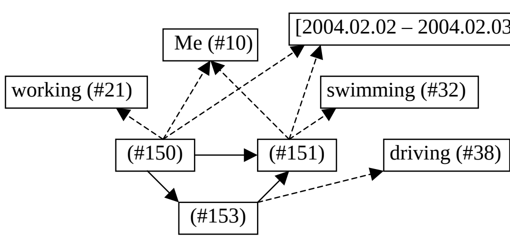

Figure 5: Incrementally augmenting the knowledge

A third example, in figure 6, suggests a way to represent a common situation: that of interacting objects. Think about the event “I was riding my horse”.

Besides the obvious stances of “me” and “my horse”, it seems natural to delimit the shape of “me on the horse”. Now it can be seen how the state of “me riding” and “my horse ridden” are included in the latter one, which makes the bridge between them.

4.4 Resources

Natural language has traditionally been considered at the core of the semantic nets. A characterization of their “semanticness”, as lying in their being used in attempts to represent the semantics of English words [2] was not really challenged in time. With our approach we shift somewhat away from this view, considering that natural language words constitute merely an important but not indispensable means facilitating access to knowledge. Words are then not represented as such in the base. A more generic re-interpretation and access methodology was the purpose that led to the separation of the knowledge network from what we call resource data.

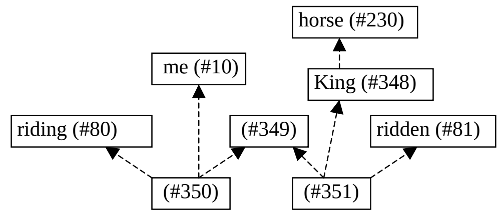

Figure 6: Example showing a case of interacting objects

In a stored stance network there is no information that may help a user remember which stance (address) corresponds to which piece of reality.

Resources are data packages (bit sequences) that can be rendered by a device into a human perceptible form. Typical resources are text, images, sound, video, etc. An algorithm for their decoding is used for each type of resource by the rendering device to render them through an output mechanism (screen, speakers, printer, etc). Since storing resources is also projected for an unlimited time, decoding algorithms should be built on open conventions that are not expected to be modified or outdated, such as Unicode for text, JPEG or PNG for images.

Storing resources could be done in any classical way, providing that each resource has a unique id, and some reference to its type, thus its decoding method. Stances can then be associated to resources to be useful for a double purpose:

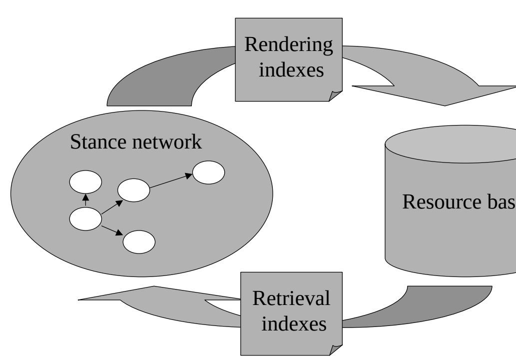

Figure 7: Associating resources with stances

First, resources act as a means to remind the user which piece of reality corresponds to a given stance. Secondly, some resources may be indexed and quickly found when needed, thus serving as reference points for accessing stances (figure 7). In particular, text resources can be indexed by alphabetical order.

Even if it is desirable to aim towards stance-resource associations that are as close as possible to one-to-one mapping, it is not a requirement. For instance, two terms that are synonyms may be connected to the same stance, and conversely it may be ok that several stances point to the same resource, such as a video, or a term with multiple meanings.

Contrarily to links in the stance network, stance-resource associations are not necessarily definitive. One may choose to replace associations with new better or more precise ones, if he finds they enhance his access to the knowledge base.

5. IMPLEMENTATION AS MEMORY AID

An implementation of our model under the form of a personal memory aid is presented in this section. In its present form, our prototype is intended to validate the feasibility and complexity of the architecture supporting the model, and its applicability for real-life scenarios.

After a first implementation for PC, we focused on a second version on PDA computers, to exploit the advantage of portability. The intention was to experiment its usage in unforeseen scenarios, collecting knowledge the same time the interesting events happen, and exploiting them as soon as they are needed. The PDA implementation was also an occasion to test accessibility and optimize for the conditions of constrained material resources (screen, input, storage and computing).

The central part of the implementation consists of an API offering higher level access functions and data structures for the management of storage memory. Access functions include stance management, resource management, and rendering and retrieval index management. In each group, there are content recording functions and content retrieval functions. For resource management we relied upon structures offered by the underlying operating system, while for the low level data structures supporting the stance network we designed a memory management system that is optimized for the typical retrieval functions taking into consideration the permanent nature of storage allocation.

Two components were then built to make use of the API: A knowledge reconstruction and navigation interface (Finder), and a visual recording interface (Editor).

5.1 Retrieving stances using the Finder

The Finder uses a simple visualization convention to display a concise overview of a stance. It is intended to be highly configurable and can include the stance ID, eventually a list of displayable resources (text, icon) and list of neighbors of a given type rendered recursively. Recursive rendering was chosen because of the observation that neighbors, especially containers, can give an instant hint to identify a stance. The role of the finder is to allow quick navigation in the stance network exploring links, and also to look up resources and locate associated stances.

In its actual state, the Finder is a minimal application handling the most common access facilities. In its main view, it displays a summary of a stance consisting of its ID, its associated text resources (if any) and one or two levels of containers, rendered recursively but without stance ID (that can be cumbersome for a quick glance upon the stance).

With the Finder, a user can locate a stance directly, or by browsing from a given stance. Locating directly can be done by entering the stance ID, or by locating an associated resource. Currently, from within the Finder it is only possible to search through text resources, and then retrieve a corresponding stance.

Browsing can start from a stance that was located directly, or from the stance that was last viewed (which is useful because often different stances taking part in the same story may already be close to each other, and found in only a few steps). The user can select one of the 3 possible link types and the link direction, to display the list of neighbors through that link type. When clicking on a stance in the neighbors list, the view is centered upon it, and the neighbors of the new stance of the current link type are displayed. Thus, walking through connections of the same type can be quick, one tap per stance.

In practice, most of the time it is enough to display containment links, switching to transformations or revisions is much less frequent. Often it is enough to see that a stance has transformation links or revision links, without actually needing to display them. Hence it proved useful to display such stances differently in easily distinguishable format.

5.2 Editing stances using the Editor

Recording new knowledge can be done in a straightforward way using the Editor interface. It appears as a stage, or workspace destined to gather stances (the actors) that will take part in the new knowledge to be recorded.

Before creating any new stance, the most important is to make sure that it does not already exist in the memory. Therefore, the Editor calls up the Finder every time a new stance is created. There, the user can browse existing stances, and create a new resource if needed. If no stance was selected in the Finder when it resumes control to the Editor, a new stance is created by the Editor. Also, any newly created resource that was returned by the Finder is connected to the current stance by the Editor. Of course, since resource association is not definitive, the Editor can also disconnect resources from stances.

Stances that are set up on the stage can then be connected through any of the 3 link types. The current link type can be selected the same way as in the Finder. Then, selecting couples of stances creates a new link of the current type between them.

Since storage of new knowledge is intended to be definitive, special care should be taken that stances and links created on the stage reflect with the highest fidelity the ideas the user wants to represent. Therefore, anything that is created on the stage is first stored in a temporary storage, that is committed into main memory only after the user reviewed them and closed the Editor.

5.3 Add-on applications

Together, the Editor and Finder offer basic access facilities that can be useful to capture simple situations. They proved good enough to start building the knowledge base. Nevertheless, they can appear insufficient when recurrent situations require repetitive actions.

We have implemented other applications on top of the core API to manage specific frequent situations, such as a People manager and a Family manager. These applications use interfaces that resemble to classical personal information management applications, where filling in fields and controls create entire sub-networks of stances. In order to properly connect to existing stances of the user, the first time they are started, add-on applications are supposed to be "set-up". During this phase, stances that represent the concepts corresponding to their interface elements should be selected from the user's stance base, or they are created if they were not yet represented.

Add-on applications use the same temporary storage to generate stances and links, as the one used by the Editor After finishing the edition, applications pass the temporary storage to the API which calls the Editor to allow the user to review and eventually complement generated stances with exceptional knowledge that are not handled by the application. This is also a way to ensure that add-on applications do not create stances the user is unaware of, or uncomfortable with.

However, we feel that this solution is just a temporary one: Such add-on applications are created in the classical manner, by programmers, who have their own ideas and views about how to stancify typical situations. We believe that any user should be free and able to create their own interfaces to handle their typical situations. Therefore it should be important to well formalize the foundations for creating add-on applications and their possible interactions, so that the user could keep control on recorded stances which should reflect exactly his views - as will be mentioned in the section about future research.

6. CONCLUSIONS AND FUTURE WORK

Admittedly, our overall ambition with this model goes beyond that of a support for personal knowledge management: to try to promote the emergence of a language. As suggested before, the success of the model depends mainly on the capacity of users to manage their personal knowledge bases, keeping them redundancy-free ensuring the uniqueness of stances. Such a social validation is the only way to evaluate the pertinence of the model.

First, people getting to know the model would wish to store their knowledge through simple applications to manage common situations. Then most of them will try to store more diverse experiences. Probably some will store too greedily and, in time, eventually lose control over their knowledge base: difficulties to retrieve and interpret their knowledge, redundancy will grow out of manageable limits. Others may find a reasonable balance between the rate of accumulating new knowledge and the ease in finding their path when retrieving existing knowledge. Some of them would discover nice, proper configurations to store particular situations. These ways of representations (formulations or expressions) will be shared among people, and spread by a form of natural selection separating the fittest configuration types. If this process continues until a number of formulations for the most current situations of everyday life will eventually gain wide acceptance, we could then talk of emergence of a language.

A solution to uphold spreading and assemble users' feedback would be to provide a reference point through a Website maintaining a collection of common application scenarios and best ways to handle them that have proven successful. From our experience, typical scenarios of personal information management, such as maintaining data about people, storing personal events, and events and relations of related people, genealogies, etc could be easily adapted to the Uniscript model. They appeared as good candidates to instigate stancification.

On the other hand, focusing too much on automatic tools proposing stancification of every experience of the subject could quickly become overwhelming for the user and lead to loss of control over his knowledge base, as mentioned earlier. It should happen progressively, directed by his confidence in mastering the content and access to his knowledge.

As for future plans, there is much work to be done, in particular to ensure the independence of stored knowledge from the storage device, implementation and synchronization between different platforms (in particular PC - PDA), access to distant resources (URI), and more advanced navigation and searching methods. Some more interesting plans concern the design of a simple interface generator, and handling uncertain stances through a storage buffer.

The interface generator would enable the user to create and personalize interfaces to manage recurring configurations. It results from the above considerations the importance that the user has control over every detail of how stances are generated, and applications do not impose him their way of structuring knowledge.

Another component, the storage buffer is useful to handle future stances and stances of uncertain source. As we argued in section of stance definition, stances that did not yet begin to exist cannot be recorded. The storage buffer could offer a means to pre-record uncertain stances that can be connected to a possible event which would allow the observer to confirm or reject their existence. For future stances such as plans, appointments of the user, this would constitute the support for a reminder mechanism in case of using as personal information management.

In conclusion, we used the perspective of representing knowledge through the viewpoint of an observer, in an attempt to challenge the stalemate in which knowledge representation seems resting today. We believe to have reached to a stable framework that takes into account the subjectivity of an observer when delimiting elements of the reality he perceives. It is not only a model, but also a methodology about how to collect and exploit personal knowledge, thus taking one more necessary step, compared to most state-of-the-art knowledge representation models, that only care about how to represent, without giving suggestions about what to represent [3]. Finally, it is clear that shared effort is needed to well prepare the bootstrap of the evolution process, which should adapt and mature in permanence within a supporting community.

7. REFERENCES

[1] J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579, 1994.

[2] R. J. Brachman. On the epistemological status of semantic networks. In N. V. Findler, editor, *Associative Networks: Representation and Use of Knowledge by Computers*, pages 3–50. Academic Press, 1979.

[3] R. Davis, H. Shrobe, and P. Szolovits. What is a knowledge representation? *AI Magazine*, 14(1):17–33, 1993.

[4] R. Dawkins. *The Selfish Gene*. Oxford University Press, 1976.

[5] P. Dourish, W. K. Edwards, and al. Extending document management systems with user-specific active properties. *ACM transactions on Information Systems*, 18(2):140–170, april 2000.

[6] E. Freeman and D. Gelernter. Lifestreams: A storage model for personal data. *ACM SIGMOD Bulletin*, 1996.

[7] J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong. Mylifebits: Fulfilling the memex vision. In *ACM Multimedia '02*, pages 235–238, Juan Les Pins, France, december 2002.

[8] D. Huynh, D. R. Karger, and D. Quan. Haystack: A platform for creating, organizing and visualizing information using rdf. In *Semantic Web Workshop*, 2002.

[9] H. J. Levesque and R. J. Brachman. *A fundamental Tradeoff in Knowledge Representation and Reasoning*, chapter I-4, pages 42–70. Morgan Kaufmann, San Francisco, CA, 1985.

[10] M. Minsky. *The Society of Mind*. Simon and Schuster, 1985.