Critique of 'Semprola: A Semiotic Programming Language'

ANTRANIG BASMAN, Raising the Floor - International, England

We supply a critique of the paper *Semprola: A Semiotic Programming Language*, suggesting directions in which its work of bringing semiotics to programming can be refined, and supplying opinions on areas where it may be refounded.

1 INTRODUCTION

Semprola: A Semiotic Programming Language by Sharpe presents a novel reframing of the task of programming languages, persistence and distributed architectures to take account of the essentially contextual nature of all symbolic expressions, not to say meaning itself.

This paper aligns strongly with our goals for the workshop in seeking to promote alternative paradigms for computation, and to establish connections with philosophical and semiotic traditions. The material in section 2.1, "Let's assume there's just one ontology" frames a pervasive and deep-rooted problem with the underpinnings of virtually all existing programming languages and all persistence technologies. It argues for the importance of allowing multiple, cooperating and related ontologies to represent a design. Software would be much less punishing to the public if developers spent more time reading material like this and thinking of ways to support such ontological plurality. Part of the reason, of course, is that it is somewhat technically difficult to do so — but no more difficult than many of the towering technical achievements of computer science. A more important part of the reason is that such work falls into an ontological "blind spot" of engineers and scientists, a problem that this paper does much to dispel.

This critique will focus on some areas where the work of bringing semiotics to programming can be refined, and give some opinions about areas where this work could be refounded.

2 PROBLEM OF FIXING AN ONTOLOGY

Linked central problems facing a successful semiotic programming (SP) system are that in most realistic contexts an ontology cannot be unambiguously fixed at the time of an author's expression, and further, that ontologies themselves could not be expected to have separable identities with universally intelligible names. In section 4.7, the author states

> ...the creation of the sign does not just depend on the content of the signifier that was captured at "author time", but also on the current content of the ontological space and user's context in which the sign will exist at "use time".

This implies that any such useful content could indeed be captured at author time — that is, that one could expect an author to be able to give any clear account of which ontology, if any, a particular sign usage should be referred to. Figure 1 shows an interesting such "confused ontology" resulting from variant usages of the term "knitting". One definition (from the OED) has it as

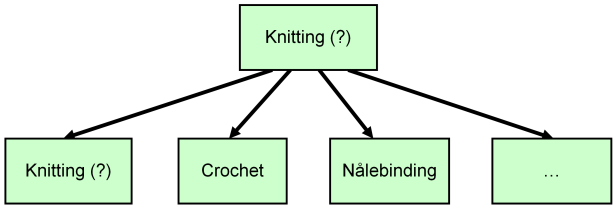

Fig. 1. A confused ontology — perhaps one ontology, or two

> The formation of a fabric by looping

whereas another (from dictionary.com) has

> To make (a garment, etc.) by looping and entwining (yarn, esp. wool) by means of long eyeless needles or by machine

The set of actions designated by one definition is strictly nested within the other. In terms of this structuring, this situation is identical with one referenced in the paper in section 3.1, that the French word "mouton" can designates what in English can be resolved as two separate cases, "sheep" and "mutton". However, this situation is inherently more problematic, because whilst a speaker of French is typically aware at the point of speaking that their words should be referred to the French ontology[1], and their hearers agree in this, the user of the signifier "knitting" may well not be conscious at the time of speaking that their speech should be attached to one ontology or another, or even that there might be more than one potential ontology in play, and furthermore it is not particularly clear from the structure of figure 1 how many ontologies there should indeed be that cover this terrain, and if there are, what names could be given to them that would result in their identities being intelligible to anyone who wished to resolve the signifier.

From the experience of this author, the fact that it is possible for expert knitters to happily sling around the term "knitting" without the slightest awareness of an ontological problem, and that they and their hearers may happily communicate on this basis for decades without confusing each other, must be counted one of the miracles of human cognition. Any successful SP system will also have to handle such situations gracefully, which implies that it must be possible to bandy signifiers indefinitely which never, from the point of view of the authors, become definitively attached to one ontological space or another. This implies the need for at least one more level of indirection in the system, beyond the one the author identifies in footnote 7, "So a simplified view of SP is that it introduces another layer of indirection from the variable to the referred to value.".

Author's address: Antranig Basman, Raising the Floor - International, London, England, amb26@ponder.org.uk.

[1]Barring strange mishaps such as that of Moliere's hero of *Le Bourgeois Gentilhomme* who had been speaking prose for more than forty years without knowing it.

3 PART-WHOLE DISTINCTION PROBLEMS

More profound ontological problems arise when different observers disagree about the boundaries and multiplicity of the activities or objects forming the referents. Still worse, if these boundaries do not nest properly.

Further text in section 4.7 reads:

> Note too that SP signs can refer to more than one object across more than one ontological space. The idea of this is that different objects might at best at representing different aspects of the real referent.

This harbours an even more profound ontological problem — that observers might even be capable of agreeing on what should be the "real referent" — or how many of it there are, where it is, where it starts and stops. Figure 2 shows an elementary version of this problem. Boundaries have been drawn separately delineating a man and a horse — which another observer may consider a single referent, "cavalry"[2]. Yet a further observer may consider that the referent of "cavalry" is the complete collection of men on horses shown. How could these different observers be put into correspondence? Some of them use the same word, and some of them use different words, but yet they are all addressing the same underlying situation, and an effective semiotic or ontological system should be able to account for the discrepancies between their view of it, and allow them to relate with one another.

This is nonetheless a simple example of this problem, where the boundaries of the objects of interest are properly nested. The natural world, for example, abounds in much more profound problems. It is interesting that the author picks as a central example of a signifier "Tree" since this apparently harmless term can lead us into the deepest thickets. Consider for example,

- The mychorrhizal networks which surround the roots of almost all trees. These may be attached to a single tree, or join collections of them. If they are undisturbed, they will be attached to the tree for its lifetime, and used to route nutrients to and from the tree to suitable targets in the environment. Should these be accounted for as a component of the tree or not?
- Suckers which grow from the base of trees. These are genetically identical to the tree and generally attached to it, but could be (and sometimes spontaneously become) detached and quickly become viable trees in their own right. Is this a situation of multiple trees or a single one?
- Trees may engage in clonal reproduction. This produces what appear to be multiple trees, but are all genetically identical, and may become widely dispersed. An extreme example is "Pando", an aspen grove in Utah which consists of 47,000 stems covering more than 100 acres, which has a root system which is now over 80,000 years old. Pando is considered "an organism", but is it more than one tree?

Further example abound in contexts of perception. Korte's law in psychophysics [3] describes the phenomenon of apparent motion of two successively presented stimuli. Under some conditions, for example, two discretely presented lights may be perceived as a single moving light. Different observers would then disagree not only about the multiplicity but also the nature of the referent of "the light". Departing from examples backed directly by physics, metaphorical or artistic uses of signifiers raise problems which just multiply from here.

Fig. 2. A confused referent — perhaps one object, or two

Whilst it is not explicitly referenced, the author's work falls in the tradition of work descended from [2] which is the root of a wide literature on techniques for "subjective" or otherwise "context-oriented" programming. An element in common between members of this tradition is that object boundaries are sacrosanct, and whilst different observers may disagree on the nature and behaviour of objects in question, they may not disagree on their multiplicity and relative arrangement. Semprola may be seen within this tradition through its reliance on SPUIDs, unique identifiers which are used to correlate whether different signifiers are bound to "the same referent". This problem is referred to in [1] as "artefact boundary intention", a kind of "excess intention" which our current means of expression force us to imbue our designs with. This is a profound problem to which this author is aware of no effective and mature solutions, but some directions towards an approach are sketched in the next section.

4 INDIRECTION THROUGH STATE

This author's opinion is that (at least) two further levels of indirection (as well as probably some changes in modelling of primitives) need to be added to produce a successful ontological system. These comprise

- The indirection through "shards" of ontologies which may be opportunistically assembled and disassembled by ad hoc communities of shared interest.

[2]This example is taken from [5]

- An indirection through the realm of "state".

These reflect a distinction of Whitehead [7]'s, between the realm of Actuals (housing what he named "Actual Entities") or "extended realm"[3], and the realm of Potentials.

By modelling the extended realm primarily as initially uninterpreted state, one can hope to establish alignment between apparently mismatched ontologies by indirection on the coordinates of this state. Naturally this forms just another ontology in itself, since there is no reason to expect different observers to agree on either the coordinates or their contents, but by making this extended realm as thinly interpreted as possible, one can hopefully push ontological disputes into the periphery of the system, as well as more easily expressing transformations or lenses mapping views or lenses on this state between observers. This implies that a problem identified by the author, that of the use of "naked data" and "unitless quantities" in section 2.3, is less pernicious than it might be. In practice, we live in a sea of unitless quantities, and by explicitly surfacing this at one level of the system design we can allow for a properly free association of quantities and other metadata with their referents.

In terms of prior art relating types and dimensions, several type systems do permit representation of rich dimensional information — e.g. the Haskell "units" package, or the F# "Units of Measure" metadata[4]. The issue, more clearly, is that it is difficult to transmit such metadata or type information in an intelligible way — especially to a system based on a different linguistic or type foundation. This author believes that such information should itself be folded back into the base representation of pure state in the system, so that it may be transmitted and processed by systems with varying levels of insight into its structure. Semprola has the required uniformity in that all state is represented in the same terms, but still too much of the interpretative machinery is bound into the protocols necessary to transmit and decode state.

5 CONFLICT RESOLUTION

The preceding sections make it clear that "ontological dispute" or disagreement needs to be not only tolerated but actively welcomed into the system in order for it to perform a harmonious and productive function in real human communities. In light of this, the short shrift given to this process in the paper is problematic. In section 5.2 is the remark that an assignment which appears to be into an ontology in which it is not permitted by the author might "throw an error condition" — but this is not really viable, from the point of view of either semiotics or usability. To start with, we have the crucial ergonomics of making sure the system as a whole never destroys user data — a pernicious property of virtually all traditional software which it has to be one of the crucial goals of SP to eradicate. Secondly, there is the issue that "setting the property of an object in a particular space" might well not be a suitable primitive for an overall SP system. A system designed around human cognitive ergonomics, as promised by its foundation on semiotics, would more appropriately allow **all** updates of state by some actor to be honoured in some arena where they will (initially) definitely succeed. These updates would then have the option to propagate to one or more linked spaces in accordance with some previously established ontological policies. This follows normal conversational practice where messages from one's fellows that appear ontologically confused or seem to be attempting an ontological overreach are not summarily rejected as being in error, but are retained perhaps in hope of future disambiguation or qualification, but at least in service of ordinary decency.

To this author, the thing called "conflict resolution" has to be positioned at the very centre of such a system, and the system should be capable of persisting indefinitely in a state where there are fundamental inconsistencies, and should not register these as "error conditions" as such — merely a metric that users of the system might care to interest themselves in from time to time, and if they are so minded, to reduce.

6 CODE AND EXECUTION

Section 2.4 weighs in effectively against the predominance of functions and the lambda calculus as the central organising unit of implementations. Indeed, the function application metaphor for computation is one of the central evils of software engineering. However, insofar as we have any "code" in a system implementation, there are worse ways of packaging it than as pure, free functions.

This issue of "what is and what is not code" is a duality which could be explored more thoroughly. Section 4.6 argues that there will be "the" SPVM which will execute its "instruction codes" but it seems far more likely that any system such as Semprola could only ever succeed through explicitly planning for a plurality of implementations, processing ontologies through variant means and at varying levels of fidelity. Expecting something as complex as an SPVM to be hosted within every language system and virtual machine in the world is probably expecting too much — and it would be most valuable if Semprola's description could separate the taxonomy of signs and public representations (e.g. agreement on what is "hot", "cold", what can be a SPUID, etc.) from any particular execution model. We look forward in a future paper to seeing the details of how the SPVM execution model works and how it might be usefully subsetted by agents who wish to implement support at some intermediate level.

7 LEAKINESS CONSIDERED BENEFICIAL

The following connection in section 5.1 seems a "loose joint":

> Indeed, as the runnable nodedges with SPVM instructions are part of the SP graph it will be possible to write such compilers in Semprola. In this kind of way the SP environment tries as hard as possible to avoid being a "leaky abstraction" where new bits would regularly need to be written in C.'

The obsession with completely homogeneous systems, and ones which if necessary could be written in themselves, is a kind of "will o' the wisp" which has led generations of language designers to produce otherwise highly worthy but ultimately unsuitable systems such as Lisp or Smalltalk. I think that designers should be more willing to entertain the idea that, whilst being perhaps a little "leaky", a heterogeneous system could be a superior one. In order to make

[3]Following Descartes, the *res extensa*

[4]described at https://en.wikibooks.org/wiki/F_Sharp_Programming/Units_of_Measure which, incidentally, cites exactly the same well-attested Mars Climate Orbiter disaster

our systems tractable from the end-user programming point of view, we need to vastly reduce their computational expressive power, and, quite certainly for example, prevent them from being Turing complete. It is terribly impressive to be able to implement a system in itself, but in the end a property which marks it as a certain failure. After all, the whole point of SP is that we expect our ontologies to be "leaky", but hopefully, ultimately, in a bounded and productive way — why not plan for our implementations to be leaky as well.

However, Section 5.1 breaks some highly valuable ground in starting to reposition the role of "language" in the overall ecology of users and authors. Whilst languages remain "single ontology" constructs, the dialogue which may be had with them will be stifling, authoritarian and limited [6]. Section 5.1 usefully starts to speculate about other roles which the things once known as languages might play in a truly pluralistic dialogue, but this section and the rest of the paper doesn't go far enough in thinking about how to disintermediate the current "package deal" of programming language, based on the primacy of source code, with the primary audience being the development tool chain. There are lots of promising threads in the paper, primarily the developing split between dead and live artefacts, and the resulting public data model, that could be developed further in order to explode this package deal.

8 FURTHER TRADITIONS

The introduction of philosophical techniques into our interaction with machines is highly welcome, but semiotics is just one of a family of techniques for assigning and assessing meaning. It can be seen as just one of a set of "analogy-forming techniques" which form various philosophical traditions. This author would particularly welcome an extension of these ontological modelling techniques into, for example, Lakoff and Johnson [4]'s tradition of meaning generated through metaphor.

9 SPECULATIONS ON PERSISTENCE

It would be valuable to investigate whether there any persistence technologies which are more or less suited to the storage and query of the vast distributed graph which SP will operate on. We should establish what properties will require to be optimised by such an SP substrate, and how these properties compare to those which the goals in the more or less mainstream persistence technology have led them. Graph databases such as Neo4J or OrientDB are aimed at related problems of storing relatively unstructured, large-scale knowledge graphs but it is unclear whether their aims align well with the needs of semiotic programming and its distributed execution.

10 CONCLUSION

Semprola breaks vital ground in bringing subjective interpretations of user data to computing, in the form of a semiotic data model and execution scheme. Further expansion of its model, in terms of dealing with everyday ambiguities that arise from problems such as unfixed ontologies and part-whole distinctions, will be necessary to turn it into an effective user tool. We eagerly await publication of further details of its data structures and the reasoning which has underpinned their development, and also the proliferation of variant subjective programming systems based not only on semiotics, but also on alternative philosophical models such as metaphors and devices drawn from critical theory.

REFERENCES

[1] Antranig Basman. 2017. If What We Made Were Real. In *Proceedings of the Psychology of Programming Interest Group.*
[2] William Harrison and Harold Ossher. 1993. Subject-Oriented Programming: A Critique of Pure Objects. 28 (01 1993), 411–428.
[3] Adolf Korte. 1915. Kinematoskopische Untersuchungen [Cinematoscopic investigations]. *Zeitschrift für Psychologie* (1915), 193–296. Issue 72.
[4] George Lakoff and Mark Johnson. 1980. *Metaphors We Live By.* University of Chicago Press.
[5] David Marr. 1982. *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information.* Freeman.
[6] Alvaro Videla. 2018. Lector in Codigo or The Role of the Reader. *Proceedings of <Programming '18> Companion, Salon des Refusés* (2018).
[7] Alfred North Whitehead. 1929. *Process and Reality.* Free Press.