Why are Interlanguages and Synchronic Computation of any relevance to you?

www.isynchronise.com.

Synchronic Computation is a style of computing based on a novel language system called interlanguage, which has highlighted serious defects in human natural languages. A new approach to language theory and computer science indicates that the historical development of human language has not been optimal for it's use as a template for mathematical and computer programming languages. More alarmingly, defects associated with human languages also place a question mark on their adequacy as efficient media for expressing rational, connected thought.

The grammar of any natural language is based on a hierarchical tree structure, whose tips are words in a language, and whose nodes represent various grammatical parts of speech such as main clause, relative clause, noun phrase, verb phrase etc.. It is argued in a report [1] published on the above website and the arXiv web archive, that tree syntax has two serious defects:

The Single Parent Restriction.

It is in the nature of the hierarchical tree form, that any part of speech may only be a subpart of at most one more complex part of speech. For example, a noun phrase may only participate in the clause or sentence in which it resides, and not in other clauses or sentences. A relative clause is only relative to the main clause of the sentence of which it is a part, and not relative in addition to any other sentences. This aspect of syntactic construction in tree languages, will be referred to as the single parent restriction (SPR).

A language system possessing SPR, limits a part of speech describing an object, from participating directly in the expression of more than one relationship, unless some form of repetition is used. As a consequence, tree languages are not ideally suited to describing many-to-many relationships between objects, which happen to be everywhere in human experience. In normal discourse, pronouns or some form of naming of complex objects is used, to ameliorate the effects of SPR. It is argued in [1] the circumventions represent partial solutions which discard an opportunity to devise a better general purpose language structure, and that conventional tree languages are wholly unsuited for describing many-to-many relationships between objects.

It is likely that SPR has been carried over into leading, semantic representations of language in human cognition, even if later processing produces semantic networks [2] not subject to the SPR, where an object node may participate in many relationships. The circumstantial evidence for this is the semantic aspect of parts of speech (e.g. noun phrases, verb phrases), suggesting trees are significantly involved on the semantic level, and the fact that a non-trivial amount of work is required to transform a tree into a non-tree network or semantic network form. If this is the case, then a further sub-optimal outcome ensues for natural languages, as well as for formal languages whose semantics preserve the tree aspect of their syntax. In the next section, it is explained how SPR places obstacles in front of the goal to arrive at a language system that conveys many meanings at the same time.

Inability to support multi-serial forms of language.

It is known that the mind is capable performing many kinds of operations at the same time, but human language seems to have an inherent serial aspect. At a cocktail party, we may attend to one conversation in the general hubbub, but not more than one. Outside of poetry or literary forms, we may not generally convey more than one basic sentence meaning at the same time in day to day communication. Multi-serial versions of computer languages can already be said to be understood in some sense, by programmers of parallel computers, who must describe and orchestrate multiple simultaneous events[1], even if the programs are initially read in a serial fashion.

Leaving aside for the moment the issue of whether we might even be able to understand a multi-serial form of a declarative language delivered in real time, it is argued in [1] that a failure to address SPR directly in designing a language system, also obscures which sentences or parts of speech can be processed or understood at the same time. Human language cannot directly express spatial information indicating information transfers and allocation of jobs to mental resources, which would be needed for multiple meanings to be understood simultaneously. Solving the SPR problem, also leads to solving the non-spatial issue in human language, potentially opening the door to the design of multi-serial forms of communication. Although our physiology is not oriented towards it, there is nothing physically impossible about multi-serial communication. If the requisite cognitive infrastructure were in place, then explicitly multi-serial communication might be achieved through the means of microwave channels, transmitted and received by physical devices attached to the brain by a neural interface.

The emergence of the non-spatial tree, as the de facto, standard language structure for humans, has had serious consequences for our capacity to describe and reason about complex objects and situations. The inability to directly share subexpressions leads to an exponential increase in the size of expressions, leading to excessive representational and computational repetition in human cognition. SPR leads to disconnected representations of environments, and a kind of linguistic schizophrenia. The serial-oriented, and non-spatial nature of language, places fundamental limits on the speed and complexity of human thought[2].

Implications for Computer Science and Mathematics.

Natural language's defects have consequences not just for our ability to communicate with each other. The defects have been carried over to computer and mathematical languages that are based on natural syntax. It is argued in [1] that tree formalisms have held back the development of high performance computers, and deterred the introduction of an explicit notion of time and computation into mathematics.

Non-spatial languages are unsuitable as parallel programming languages for multi-core computer architectures, because their compilation and efficient mapping to resources requires the solution of a combinatorial explosion, which is argued to be one source of the parallel computing crisis. The use of names of constructions, or pointers to locations in a

---

[1] Whilst music (without lyrics) does not explicitly convey declarative information, it's composition does require the appreciation, description and scheduling of sounds from simultaneously active instruments.

[2] It is open to question if the limitations of natural language contribute to irrationality and aggression in human behaviour.

computer memory alone, to access and reuse results of subexpression evaluations, represent a partial solution which discards an opportunity to devise a better general purpose computer language structure.

The problematic nature of subexpression repetition in tree languages has been noticed before, and has given rise to graph/data flow models, such as Term Graph Rewriting, Petri Nets, Semantic Nets, and Dataflow models. But these approaches have not entered the mainstream. Although the basic structure used is that of a graph, they are described in conventional tree-based mathematics, involving the non-spatial transformation of expressions alone, and lack an explicit notion of a computational environment. They are implemented on networks of Turing Machines/processors, do not call for a fundamental rethink of formal models of computation, and rarely call for an alternative computer architecture to the processor network. The next two sections discuss these matters further, and are more technical in nature.

Interlanguages and Synchronic Models of Computation.

An alternative to conventional tree based syntax and semantics has been devised in the form of an a language environment called *interlanguage*. The environment consists of a language based on the notion of the interstring, and an abstract memory and functional unit array, capable of storing elements, and performing operations of some given algebra. Interlanguage allows the sharing of subexpressions to be explicitly represented, with linear cost with respect to the number of subexpressions. The tree form of an interstring is highly regular, requiring only a minimal syntactic analysis phase before semantic processing. Interstrings indicate which subexpressions may be semantically processed simultaneously, and allow resource allocation to be performed implicitly. Interstrings are also suitable for representing data structures with shared parts, and are intended to replace trees and graphs as standard programming structures.

The α-Ram family of machines are formal models of computation, which have been developed to be the target machines for the compilation of high level programs expressed in an interlanguage. Members of the α-Ram family with infinite memories are Turing Computable. A member of the α-Ram family with finite memory, called the Synchronic A-Ram, may be viewed as a formal model underpinning the concept of an FPGA or reconfigurable machine. It supersedes finitistic versions of the Turing Machine and the l-Calculus, the current standard models of Computer Science, in it's ability to efficiently support a high level parallel language. There is the prospect of a proper formalisation of parallel algorithmics, a new way of relating operational and denotational program semantics, and novel opportunities for parallel program verification. Massive instruction level parallelism can be supported, storage and processing resources are integrated at the lowest level, with a control mechanism similar to a safe Petri Net marking.

An interlanguage called *Space*, has been designed to run on the Synchronic A-Ram. Space is an easy to understand, fully general purpose parallel programming model, which shields the programmer from low level resource allocation and scheduling issues. Programs are textual rather than graphic, and iteration, data structures, and performance evaluation are supported. Space has a high level sequential state transition semantics, and solves the conceptual problem of how to orchestrate general purpose parallel computation, in a way that has not been achieved before.

The set-theoretical/logical definition of procedures for assembling constructions in mathematics, and the constructions themselves, are normally considered to reside in a

universe of discourse, which is neutral and abstract from any computational implementation. A claim is made however, that conventional tree based formalisms in pure mathematics, harbour implicit notions of sequential, asynchronous and recursion oriented computation. Further, a universe of discourse incorporating an explicit parallel computational environment, is amenable to the adoption of parallel forms of reasoning, that would bypass an implicitly sequential.style in conventional mathematical discourse.

Synchronic Engines are physical architectural models derived from the Synchronic A-Ram and Space, and are composed of large arrays of fully, or extensively connected storage and processing elements. The models suggest optoelectronic, and spin-wave based hardware specifications. If interconnect issues can be overcome, there is a new avenue for developing programmable and efficient high performance architectures.

Why the next generation of computer architectures called multi-cores, represent a dead end.

The parallel computing crisis is still with us because the multi-threading/multi-core approach has not been made to work well over four decades, despite all the current industry focus. Isynchronise is a research organisation based in London which disseminates public domain information and software, and published a report on 25.05.10, describing a new approach to high performance computing, called Synchronic Computation.

On a thread entitled Where do you see the next innovation coming from in high performance computing?, on the linkedin group (High Performance Computing (HPC).,), the Director of Intel Research at Santa Clara, John Gustafson called for a fundamental new approach to parallelism that tackles data management issues and the serial aspect of programming languages.

In a reply, Alex Berka of Isynchronise posted a summary of a research report that addressed these concerns at a fundamental level. Introductory and full versions of the report are now available from http://arxiv.org/pdf/1005.4798 and http://arxiv.org/pdf/1005.5183 respectively. The Synchronic Computation Group has been formed, and has attracted leading academics and professionals, including John Gustafson himself. (Membership naturally does not imply agreement with the Synchronic Computation approach).

Alex Berka is a researcher who wishes to place the work in his report [1] in the public domain. Summarising the report, Berka said quote: My goal was to subject the foundations of language to a level of rigour and scrutiny commensurate with Anglo-Saxon tradition of Analytical Philosophy, with a view to developing non-serial forms of language, and new approaches to foundational informatics and high performance computing.

The research is now under study by leading academics, and has fundamental implications for mathematics as well as computer science. The worlds leading forum on foundations of mathematics (http://cs.nyu.edu/mailman/listinfo/fom) accepted and posted a summary three days ago. (Posting does not imply the groups agreement with summary contents.)

Summary of Synchronic Computation

Natural Language has major structural defects considered from a parallel computational point of view, that have been carried over to programming languages that are based on natural syntax. [Insert your favourite multi-threading language] is such a language, and does not support the following features:

-An ability to directly share subexpressions in data structures and dataflow, leading to excessive representational and computational repetition, and contributing to the phenomenon of code bloat.
-An ability to tag data transfers with soft machine connections expressed at the preceding level of abstraction.
-An ability to tag operations with soft machine resources expressed at the preceding level of abstraction.

These factors are intimately related to multi-threading issues such as deadlock and resource conflict, decreasing speedup with increasing thread/core count, lack of portability, and the parallel computing crisis itself. Superior approaches for parallel computing are GPUs/SIMD programming, FPGAs, and Digital Signal Processing on ALU arrays, but all are special purpose, and suffer from not being a general purpose programming and machine model of parallelism.

The Synchronic Alternative.

Tagging and subexpression sharing in the revolutionary Interlanguage system, allow multi-threading issues to be circumvented by resolving them explicitly at the modular level. Contention issues at lower levels of abstraction have been resolved by earlier submodule composition. Synchronic Engine architectures are in design stage, derived from a brand new theoretical model called the Synchronic A-Ram, based on the Interlanguage system. Interlanguage programming tools support the above language features, and are currently in planning stage for commercially available, reconfigurable machine architectures. Higher performance is expected from dedicated Synchronic architectures, that together with Interlanguage toolkits will enable

- faster processing
- simpler management of complexity
- better software verification and maintainability
- better portability

Initial studies suggest Synchronic Engines/Interlanguages represent a general purpose solution, that could offer up to 100x speedup on conventional platforms.

Interlogic Database Models.

The mathematical basis of synchronic computation suggests a family of new database models called Interlogic Database Models, oriented to massively parallel query processing and transaction updates on Synchronic Engines and compatible reconfigurable architectures. Novel Customer Relationship Management applications are in planning stage, that will exhibit enhanced deductive capabilities and processing volumes. Further information on development timeframes, and compatibilities with industry leading CRM tools will be available soon.

Why Now ?

Optoelectronic, molecular, and spintronic technologies are now under development, that have outgrown the basis of current programming models. Synchronic Engines and Interlanguages are general purpose, and bypass silicon based constraints and evolutionary baggage from human languages.

[1] A V Berka “Interlanguages and Synchronic Models of Computation.” http://arxiv.org/pdf/1005.5183 Published by Isynchronise Ltd on www.isynchronise.com 25th May 2010.
[2] Woods, William A. (1975) "What's in a link: foundations for semantic networks," in D. G. Bobrow & A. Collins, eds. (1975) Representation and Understanding, Academic Press, New York, pp. 35-82.