

TOWARD FOUNDATIONS FOR THE UNDERSTANDING OF TYPE

Douglas T. Ross
SofTech, Inc.

I have now written, have available from SofTech, and have submitted for publication elsewhere, the first two of a series of deep philosophical papers [1, 2] about Plex [3]. Among other things, I believe these papers begin to provide a proper foundation for our endeavors in computer science and software engineering. The trend of these developments is indicated in the following, which I call The Proem of Plex1 and Plex2 ["Proem: 1.) A preliminary discourse to a longer piece of writing. 2.) An introductory comment before a speech. 3.) Something that opens or begins." -- Webster's].

THE PROEM

Form couples space and time.
Only every object has form.
Type is prescription of form.
Only every thing has both object and type.
Entelechy is that which a thing is by virtue of its form.
Object is that which a thing is.
Operator is form of operation.
An operation is a thing.
Only every happening has both operation and type.
Every object is unique and has both name and site.
A class is the collective class of objects of its type.
Form is class of sites.
An object is created by decomposition.
An object is composed of its components.
A component has name, type, site, and value, all of which compose its object.
The site of a component is the site of its object.
The value of a component is that perceived of an object by its type in the context of its site.

Perception is composed of acception, decomposition, and conception.
Perception is a function of the state of the component for which value is being perceived, and of the state of the object being perceived.
Acception is composed of accessing and accepting, both of which only are functions of entelechy.
Only every thing accessed is accepted.
Decomposition is a function of only accepted entelechy.
A perception is a thing.
Conception is a function of the state of perception and is created from it by creation of a new type.
Only every thing that is perceived is a component of a perception.
Type is composed of sameness and change.
Sameness is composed of static, specific, and generic sameness.
Only every value of every component may be changed.
Change is not caused, it is allowed by change of perception.
The state of an object is a function of its perceived type and value.
A relation is a true predicate on things.
An event is a true predicate on happenings.
Sameness is maintained by allowed change composed of events and relations of state.
The highest percept is existence.

OBJECT-ORIENTED LANGUAGE

Even in 100 pages, Plex1 and Plex2 provide only some basic discussion of the first nine lines of the Proem, so interpretation is not intended to be obvious! But the words, even when read poetically, do show what we are up against in questions of "data abstraction and type". In Plex1 I say, "I, for example, would like to be able to understand,

extend, rigorize, ensure against circularity and incompleteness, etc., The Proem (which at this point should be read like poetry -- just for the impact and effect of the words, one upon the other) ... It is a summary of some of the more solid conjectures of mine regarding the "object-oriented language" aspects of Plex. Missing from the list are many of the common terms such as "variable", "assignment", "operator", etc., of computer languages and logic because their concepts are still buried deep in the above ... The list later will have statements such as:

Value of an object with respect to an operator results from the projection of the entelechy of that object on the reference frame of that operator. The value is perceived in the context of the state of that operator and is understood ..."

The term "object-oriented language" comes from the following kind of statement, which also will appear in the future:

You don't assign values to variables, you assign variables to objects.

"My hope is that the object-oriented view will have an impact on language definition and use. With its attention naturally focussed on the lifetime of an object from creation to dissolution, whether and when to copy is a central focus, not a sometimes side effect; in defining new "user-defined types" it is not the operations on but the operators of the type that are important; the distinction between data, structure, and datastructure becomes clear in terms of data types (where data are space-time objects), structure types, and activity types; "sugar-free" syntax matches closely to semantics, because abstraction is inherent, as are efficiency, reliability, and security. Once we understand better the idea of object-oriented language, our formal languages will become natural languages, not artificial languages.

SAMENESS

I take as the starting point for Plex, a very primitive derivation of static, specific, and generic sameness, based on the axiomatically-assumed "mystery" that the world is merely space-time flux of form.

static sameness is fluctuation that is ignored
(existence; mere awareness)

specific sameness is fluctuation that is allowed
(change; leaving a thing still the same unique thing, merely changed)

generic sameness is fluctuation that is typical
(transformation; asserting that different things are of the same type, merely transformed)

By the end of Plex2, I begin to exhibit some of the fundamentals of a Calculus of Type -- primarily focussing on the statement above that it is the operators of a type, not the operations on objects of that type that is the central issue. This follows immediately from the fact that the operators of a type are the basis for the prescription of form which defines that type, and any real-world fluctuation which conforms to the form (not "are formed" -- they form themselves because they possess the form they possess. [To be or not to be is not even an option!]) is a thing of that type. The way that this happens is illustrated by the following quote from Plex2 (originally from Tarski's Ph. D. thesis, quoted from Jordan [4]):

$$(p,q) :: p.q \equiv :. (f) :. p \equiv : (r) . p \equiv f(r) . \equiv . (r) . q \equiv f(r)$$

$$[i. e. , \forall p \forall q (p.q \equiv (\forall f(p \equiv (\forall r(p \equiv f(r)) \equiv \forall r(q \equiv f(r))))))]$$

"The right-hand side of the equivalence can be regarded as the definition of the sign of conjunction by means of the equivalence sign [a kind of sameness] and the universal quantifier. By adding the sign of conjunction to those just mentioned, the signs of the remaining logical constants can be defined by means of the theorems:

$$(p) :: \sim p . \equiv : p \equiv . (q) . q$$

$$[i. e. , \forall p((\sim p) \equiv (p \equiv \forall q(q)))]$$

$$(p,q) :: p \supset q . \equiv : p \equiv . p.q$$

$$[i. e. , \forall p \forall q((p \supset q) \equiv (p \equiv (p.q)))]$$

$$(p,q) : p \vee q . \equiv . \sim p \supset q$$

$$[i. e. , \forall p \forall q((p \vee q) \equiv (\sim p \supset q))] "$$

As I say in Plex2, "as long as the relation of p and q to all of the other elements of their universe (r), and to all of the possible functions of their universe (f), is such that the right-hand side of the theorem is true ... then that unique object p.q exists and is of the type conjunction." "A thing exists as long as its form persists" (from Plex1).

EXISTENCE

The crux of the matter must include resolution of the following question (from Plex1):

How can something
 be
 and be only itself
 while also being only its parts
 which also are only themselves
 while also
 it also is any other parts
 which it also only is
 which also are only themselves?

The answer seems to lie in the system of mereology of the Polish mathematical-logician Lesniewski, resulting from the single axiom (due to Lejewski [5], see Plex1):

Axiom A

$[AB] :: A \in \text{el}(B) . \equiv :: B \in B :: [Ca] :: [D] :: D \in C . \equiv : [E] : E \in a . \supset .$
 $E \in \text{el}(D) : [E] : E \in \text{el}(D) . \supset . [\exists FG] . F \in a . G \in \text{el}(F) . G \in \text{el}(E) ::$
 $B \in \text{el}(B) . B \in a . \supset . A \in \text{el}(C)$

which reads as follows:

Axiom A

for all A and all B	
A is an element of B if and only if	... this is the definition of <u>el</u>
B is the sole B and	... i. e., B exists
for all C and all a	... think of C as the class of
for all D	things with attribute a
D is C if and only if	... define D as the class of
for all E, if E is a then	a's iff
E is an element of D and	... D has all a things as elements
for all E, if E is an element of D	... and, furthermore,
then there exists F and exists G	each element overlaps
such that F is a and G is an	some a thing, F
element of both E and F	... end definition
and if B is an element of B and B is a	... if B is an element of itself
then A is an element of C	and is a, then A is an
	element of C

THE CHALLENGE

There is much work to be done to bridge the gap between these philosophical foundations and the specific efforts of on-going work covered by the other panelists. I believe the philosophy can show why the good parts are good, as well as showing where other aspects are subtly, but seriously and perhaps dangerously, in error. I expect my views to provoke stormy controversy as well as worthwhile discussion. But I hope that thoughtful colleagues will make the considerable intellectual investment needed to penetrate the smoke and help attack the fire!

[1] Ross, D. T., "Plex1: Sameness and the Need for Rigor", Document 9031-1.1, November 1975, 56 pp., SofTech, Inc., 460 Totten Pond Road, Waltham, MA 02154.

[2] Ross, D. T., "Plex2: Sameness and Type", SofTech Document 9031-2.0, December 1975, 49 pp.

[3] Ross, D. T., "A Generalized Technique for Symbol Manipulation and Numerical Calculation", CACM, Vol. 4, No. 3, March 1961, pp 147-150.

[4] Jordan, Z. A., "The Development of Mathematical Logic in Poland Between the Two Wars" (1939) in Polish Logic 1920-1939, S. McCall, Ed. Oxford at the Clarendon Press, 1967.

[5] Lejewski, Cz., "A Note on a Problem Concerning the Axiomatic Foundations of Mereology", Notre Dame Journal of Formal Logic, Vol. 4, No. 2, April 1963, pp 135-139.