Tradeoffs in Future Proofing Notations

Luke Church

Computer Laboratory
Cambridge University
luke@church.name

Keywords: POP-I.A. Social Notations; POP-III.A. Notation; POP-III.C Cognitive Dimensions;

Abstract

Designing future proof notations requires a balance of trade-offs. Vagueness and extensibility result in poor interoperability and rapid exhaustion of the notation space due to user capture. Precision results in premature commitment problems in the face of changing requirements. Viewing notation space as a precious resource may allow a more systematic consideration of the implicit trade-offs.

Vagueness and Extensibility

In his keynote speech at PPIG Green (2006) argued that notational designers were in danger of being 'seduced by the now', he suggested a possible solution is for notational designers to be as vague as possible to allow for future uses and maintenance of consistency.

However vague notations can cause problems. Notations that are ambiguously defined are notoriously difficult to interoperate with, both across organisations, and across time. A topical example is the modification of the undocumented functionality of the Windows NT Kernel, with the release of Windows Vista, breaking several 3rd party Anti-virus products (Jones 2006)

It is now a received wisdom of computer science that programming languages should be unambiguously described, despite few commercially used languages achieving the aim.

Secondary Notation Capture and Under Specified Domains

Under-specification of notations is often used as a method of providing *Secondary Notations* (Green and Blackwell 1998). For example a user will often define a folder structure that aligns with their understanding of their documents. However, vagueness or large undefined regions in the definition of notations often releases too much notational space for use in *Secondary Notation.* This is captured by users for their own purposes, turning it into their own primary notation in a different domain. In our example the folder names have been captured as a primary notation, it has syntactic meaning to the user, but not to the system.

The difficulty with this is that allocating space for secondary notations is a one way process, if it was reclaimed as a primary notation for the system, it would disrupt a user's mapping in another domain. Consider a change to the operating system that gave different meanings to your document folders based on the names they'd been given them! Therefore provision of secondary notation will not, by itself, future proof a notation.

Many standards allocate areas of notational space for custom extensions of the notation. For example the JPEG/JFIF standard allows data to be embedded in 'APP' fields. This allows cameras to write metadata such as date/time, shutter speed, etc. into the JPEG files.

Again, this is no panacea. For example the Border Gateway Protocol (BGP)(Rekhter et al 2006) contains a large amount of extensibility. However leaving so much behaviour unspecified, has resulted in a situation where it is not sufficient for a system to implement BGP, a router must also

support a set of weakly defined extensions. The effect of this is there are few, if any, complete implementations (Jon Crowcroft, personal communications). Such flexibility was added to provide future proofing, but it has resulted in the standard degenerating into 'the implementation is the standard' that it was intended to avoid.

The Risks of Precision

Vague notations cause problems in many circumstances, however precisely defined notations suffer from *Premature Commitment*. Precision in the definition of a system often results in inflexibility in the face of future changes in the use or notation or changes in the environment in which the notation operates. For example IPv4, the widely used version of the Internet Protocol, allocates 32 bits of address space. However this has proved to be inadequate but its replacement, IPv6, which provides 128 bits is proving difficult to deploy due in part to the large IPv4 installed hardware base.

Precisely specifying a notation to avoid providing *Secondary Notation* has a negative effect on usability of the system during modification (Green and Blackwell 1998). This confirms the duality of *Secondary Notation* as a Cognitive Dimension*;* it is neither an intrinsically good or bad property.

Systems that are highly defined tend to be difficult to scale. At best, small spaces of the notation that weren't defined tend to be heavily functionally overloaded, frequently with substantial technical ingenuity, to provide additional needed notation space and functionality.

For example, the exhaustion of the IPv4 address space promoted the development of NAT (Network Address Translation), which breaks some of the semantics of IPv4 addressing by mapping an entire network to a single IP address allowing reuse of much of the address space. But such trade-offs are rarely without side-effects; in NAT it breaks the link between IP addresses and machines.

Notation Space as a Resource

If neither of the extreme points of the spectrum is ideal, is there a happy medium that can be found?

If notation space is considered as a resource, network economics can be used to discuss the tradeoffs between evolutionary change (consumption of unused notation space) and revolutionary change (replacing the notation, possibly after all the space has been consumed). (Shapiro and Varian, 1998)

However, the provision of notation space is rarely the only consideration. Considering the address scarcity in IPv4 the evolutionary solution, NAT, is much more widely used than the revolutionary solution, IPv6, due to the relative implementation costs. In the short term at least, the expense of undoing the 'Seduction by the Now' with a clean solution has proven too high.

We can consider various trade-offs in the use of the notation space:

- Defined space: Space whose use is fully defined and used to provide functionality (e.g. keywords in a programming language)
- Secondary Notation space: The system will never use this space, the user is safe to capture it to define their own notations in another domain (e.g. variable names in most languages)
- Reserved space: Notational space whose use currently not permitted, to be used by the designer to meet future needs (e.g. reserved keywords that are not currently used)

The tradeoff between the allocated spaces is a complex one. The system must be powerful enough and flexible enough for today's needs, contain enough secondary notation space to be usable and maintain enough reserved space to allow for future proofing.

The provision of extra notation space often comes at a substantial cost, e.g. 128 bit addresses are technically much more costly to implement, causing further delays in deployment of the technology (Ensoft, personal communications)

Acknowledgements

Many thanks to Thomas Green, Darren Edge and Alan Blackwell for many interesting discussions.

References

Jones, J. (2006) Windows Vista x64 Security – Patchguard
http://blogs.technet.com/security/archive/2006/08/12/446104.aspx

Green, T.R.G., Blackwell, A.F. (1998) Cognitive Dimensions of Information Artefacts: a tutorial

Green, T. (2006) Seduction by the Now, PPIG 2006

Rekhter, Y. (2006) A Border Gateway Protocol 4 (BGP–4) RFC 4271

Shapiro, C., Varian H. (1998) Infromation Rules: A Strategic Guide to the Network Economy, Harvard Business School Press