Tom Gruber in AIS SIGSEMIS Bulletin!!!

"Every ontology is a treaty – a social agreement – among people with some common motive in sharing."

Tom Gruber
Co-founder and Chief Technical Officer of Intraspect Software
Former Chief Architect, Collaboration and Portal Products, Vignette
http://tomgruber.org/

1. ***Miltiadis: Tom we are delighted you agreed to this interview. Let start by asking you to provide me your general idea on how ontology-driven information systems are going?***

Information technology is like the ocean: a lot of interesting creatures evolve to fill niches, eat each other, cloud the view with waste matte, and die. All this organic material falls to the floor, where it becomes sediment and eventually the bedrock of the future. Once things have settled for a long time, industry becomes dependent on the oily mess at the bottom and everyone forgets where it came from.

We have lived through this dynamic with IT and ontologies. There has been a lot of waste matter and clouded vision, yet most new large-scale software or information systems today start with some kind of ontological commitment. The modern large-scale server software today is based on XML for interconnect and the relational database as the back tier. These are now layers of the foundation. These layers have insulated people from low-level concerns, and allowed them to accept that it's a good idea to specify the conceptual foundations of their systems at a level above the syntactic.

2. ***Miltiadis: Tom your name is synonymous to Ontologies. Don't laugh but in a way you are the Ontology Man. One of your articles has been cited many times to refer to a popular definition of the term ontology. In retrospective, will you change today that definition in some direction or in your opinion it still remains valid?***

Well, the most important components of that definition of ontology are that the ontology is representation artifact (a specification), distinct from the world it models, and that it is a *designed* artifact, built for a purpose. I think most computer scientists get the distinction between a specification and the world, even for synthetic worlds. In retrospect, I would not change the definition but I would try to emphasize that we *design* ontologies. The consequence of that view is that we can apply engineering discipline in their design and evaluation. If ontologies are engineered things, then we don't have to worry so much about whether they are *right* and get on with the business of building them to do something useful. We can design them to meet functional objectives and constraints. We can build tools that help us manage them and test them. And we can have multiple ontologies that coordinate or compete based on objective criteria rather than brand or authority.

3. ***Miltiadis: In a recent talk, you discussed the concepts of informal, semi-formal and formal ontologies. Others (e.g., Prof. Sheth) has picked up on this and cited this perspective. Can you share some insight on these? In particular, how would you consider the success or lack of success of "formal ontologies"? And what has your work on Intraspect tell us about the role of informal ontologies. Finally, is it true semi-formal ontologies represent a sweet spot? If so why?***

This question touches several important issues so I hope you will bear with me on the length of this response.

Ontologies as specifications are always a mix of formal and informal parts. The informal parts of a spec help explain something to humans, and the formal parts allow some automated analysis. For example, a dictionary is a set of terms with informal definitions. The textual definitions in dictionaries are informal because they are in free form natural language and are therefore vague, ambiguous, and context dependent. The formal parts of a specification are statements that can be used to deduce or enforce meanings. Axioms in a logical theory are formal; so are the equations in a physics theory. The term "Semiformal Ontology" refers to a ontology which has a few bits of formality but is largely informal. It is the analog of what Tom Malone calls semi structured data, such as email or office forms. A semiformal ontology could support technology that automates processing of its formal parts but leaves it to the reader make sense of the informal parts. For example, there is a semiformal ontology for cultural materials designed by a group called CIDOC. It allows libraries and museums to exchange reference data about their collections. However, it does not try to represent the content of the collections. The CIDOC ontology is not purely informal because it puts some formal constraints on the use of its vocabulary, and also it is not mostly formal because most of the vocabulary is defined in natural language.

> ***The term "Semiformal Ontology" refers to a ontology which has a few bits of formality but is largely informal. It is the analog of what Tom Malone calls semistructured data, such as email or office forms. A semiformal ontology could support technology to processing of its formal parts but leaves it to the reader make sense of the informal parts.***

This may seem muddled because there is no clear distinction between formal, informal, and semiformal ontologies. I think the "term semiformal ontologies" is more useful as a label than a definition, because every ontology is a mix of formal and informal parts. The label tells us to consider both parts when building ontologies to be of some practical use. This may make more sense in a historical context.

At Stanford we tried to get to the bottom of ontologies, so to speak, and explored foundational concepts for things of importance to software and information systems (sets, relations, quantities, units, time, etc.). For these conceptualizations we could build nice, rich axiomatized theories. In the same lab we also dabbled in fancy user interfaces, particularly the sort that led to today's dynamic web interactions (we called them virtual documents at the time). It struck me that in both areas we got technical leverage from making ontological commitments. For instance, while the ontologies for engineering mathematics were rich with constraints that would allow modeling tools to work (e.g., they could assume unit independence), the dynamic web interfaces also made ontological commitments that enabled tools.

In fact, the World Wide Web is based on a semiformal ontology, and it shows how ontological commitment works in software interoperability. At its core, the concept of the hyperlink is based on an ontological commitment to *object identity.* In order to hyperlink to an object requires that there be a stable notion of object and that its identity doesn't depend on context (which page I am on now, or time, or who I am). Most of the machinery of the early Web standards are specifications of what can be an object with identity, and how to identify it independently of context. These standards documents serve as ontologies – specifications of the concepts that you need to commit to if you want to play fairly on the Web. If one built a system with these commitments, all of the web infrastructure works well. If you violate the spirit of the ontology – such as the agreement on identity – then things don't work so well. For example, early web servers often packed a lot of state into the URLs, which violated the notion of object identity. Systems built this way could not be searched, bookmarked, or mentioned in email messages. I think that there were design weaknesses in the ontologies – ambiguities in the standards documents – that allowed *formal compatibility* with the Web without a commitment to the conceptualization on which it is based.

So, to conclude, I would say that all practical ontologies are semiformal, and the "sweet spot" is an ontology that specifies clearly how you can commit to it. Both the formal and informal parts should be designed to make it easy to play by the rules: the formal by automated testing and the informal by well written documentation.

See also http://tomgruber.org/writing/cidoc-ontology.htm

" To conclude I would say that all practical ontologies are semiformal, and the "sweet spot" is an ontology that specifies clearly how you can commit to it. Both the formal and informal parts should be designed to make it easy to play by the rules: the formal by automated testing and the informal by well-written documentation".

4. ***Miltiadis: Tom in my naïve understanding Ontologies refer to conceptualizations and engineering. What is your opinion about current ontology engineering methods? Have they evolved significantly in the last years? Which method(s) would you mention as the most outstanding?***

I am not up to speed on the latest methods and tools, but it seems to me that ontological engineering faces the same problems as software engineering. I would look for ontological engineering methods that address the issues of functional specification (what the ontology is *for*), design documentation and review, enforcing constraints while editing, unit and integration testing, and the distributed collaboration that would allow the ontologies to be developed in an open-source style.

5. ***Miltiadis: Many people in IS community are interested in ontologies. And of course one critical point towards the understanding of ontologies is the use of ontologies editing tools. In your opinion, are current ontology editing tools simple and easy to use enough for non-technical people? Or perhaps ontology edition would always be a matter of expert knowledge engineers?***

Again, I haven't reviewed the current tools, so I would be remiss to make any kind of judgment. It is possible to make editing tools that automate the low level bookkeeping and constrain checking. These tools are critical to making progress in ontological engineering; just as good language specific editors and integrated compilers were important to making progress in normal software engineering. Whether tools of this class can enable non-technical people to make ontologies depends entirely on what they are for (the tools, not the people :-). If the ontologies are to enable software interoperability of reasonably complex programs, I would say that it requires highly technical people – trained in architecture and systems thinking – to be involved. It takes experience and intelligence to anticipate how a set of programs, which have yet to be written, might interoperate and exchange information. At the same time, "non-technical people" must also be involved in an ontology design process, if they hold the vision about what these interoperating programs are going to do for us.

"Whether tools of this class can enable non-technical people to make ontologies depends entirely on what they are for (the tools, not the people :-). "

6. ***Miltiadis: Many people agree that we have to put enormous effort on the development of ontologies for domains in order to gain all the benefits from the ontological engineering. What is your view in the quality and quantity of current ontologies? Do we have enough and good enough ones, at least for the most common Information System domains? Or there is still much work to do to obtain high quality and usable shared domain ontologies?***

With all due respect, I think this sounds like asking whether we have the right "quality and quantity" of standards documents for the Web. It isn't a matter of "enough". What matters is to create agreements that enable people to make systems with ontological commitments that are useful. The bloggers and news aggregators get a lot of mileage from the RSS ontology, even though it is minimal. The term called "title" in RSS is really useful for calling out the title of an article, even if the content of the title field are not tagged with semantic clues. Do we need an ontology of the world to aggregate the world news? A lot of recent investment is betting that the answer is "no". However, for applications where we need very high precision or recall in our queries, we will need agreements on how to mark content to make useful distinctions in machine readable formats. Ontologies are good for making these agreements.

7. ***Miltiadis: I really liked a paragraph in Intraspect's mission statement I found on web: Intraspect applications help people collaborate in large distributed communities, learn from each other, and continuously contribute to a collective body of knowledge. How can we exploit all this tacit knowledge Tom? Can we go beyond verbalisms?***

This is the holy grail of the knowledge management world, and there are many ways to approach the problem. Intraspect was designed on the assumption that it is more valuable to get evidence of human knowledge into a *collective memory* than to add structure to existing online material. So we created technology that helped people work together on-line, and as a byproduct their work became available for discovery using information retrieval technology. The idea works when there is a compelling reason for people to collaborate and the technology makes it easier for them to collaborate on line in a common space than to work offline.

l
n the
content, even if the
content is largely

Although there is plenty of "tacit knowledge" that is not captured by on-line work, intelligent people can learn a lot from seeing what others have done. I was impressed by how far users of Intraspect could go by sharing a collective memory of unstructured and semistructured content. Thousands of people were able to learn from each other every day, without having to know who needed to know or whom to ask. A key insight to why it works so well is the role of *context*. With Intraspect, information was captured in the work context of its creation (projects, sales deals, customer relationships). The context itself provides a powerful semantic label on the content, even if the content is largely unstructured. For instance, if someone on a project sent an email to a client discussing requirements (and this email is captured by Intraspect), the context of the project and the client can be used to automatically tag the content with the formally enforced identities of the project and the client. Then, when someone later searches for this kind of content, they can use formal constraints on the query to slice and dice by project and client. Or, conversely, if someone discovers the email message by text search, the formal attributes of the project and client identities are available for the searcher to use in making sense of the results.

8. ***Miltiadis: Dear Tom, many people agree that a key "battlefield" for the promotion of ontologies as well as semantic web is the "realization" of advanced knowledge portal. Do you agree with this statement? What do we have to expect from the Knowledge Portals of the next generation?***

I would ask what the Knowledge Portal is *for.* If the goal of a knowledge portal is to provide answers to common questions, for instance, then I would compare the cost and benefits of creating an enforceable categorization of possible questions and answers (a Semantic Web approach) with distributed collaborative approaches such as answer gardens or the open directory or the Wikipedia. These collaborative approaches optimize for content creation over content structure, but they use social feedback mechanisms to informally validate the content. A similar approach could be used to intermediate the content in a knowledge portal (people collaborating on the categorization in the portal). In a collaborative, social environment, the role of the ontology is to help *people* communicate their intent and to make agreements.

9. ***Miltiadis: Dear Tom thank you for your time. It was an excellent talk. Would you like to give parting thoughts to our readers?***

With all of our hard work to get things right, I find it critical to remember that every ontology is a treaty – a social agreement – among people with some common motive in sharing. What we need the most are ontologies that unleash the potential of the world's population to make useful contributions to our collective knowledge.

ry
– a
eement – among
people with some
common motive in
sharing"

Thanks for your interest. For more information, see tomgruber.org.

Short Bio

Tom Gruber (http://tomgruber.org/) is an innovator in technologies that extend human intelligence. Building on early work in computer-based learning and artificial intelligence, he now focuses on creating environments for organizational intelligence. He is co-founder and CTO of Intraspect Software, which creates environments for people working together on line. Intraspect applications help people collaborate in large distributed communities, learn from each other, and continuously contribute to a collective body of knowledge. Intraspect is used by hundreds of corporate customers in Financial Services, Marketing Services, Professional Services, High Technology, and other globally distributed enterprises.

At Stanford University in the early 1990's, Gruber was a pioneer in the use of the Web for knowledge sharing and collaboration. He established the DARPA Knowledge Sharing Library, a web-based public exchange for ontologies, software, and knowledge bases. Gruber also led the Stanford team that invented and deployed the first Virtual Document applications on the web that generate natural language explanations in response to questions.

With colleagues at Stanford, Xerox PARC, and SRI, he designed systems that provide shared virtual spaces for collaborative work, agent-based collaborative engineering, and collaborative learning. To support the collaborations of the WWW research community, Gruber created HyperMail. HyperMail turns ordinary electronic mail into a web-based organizational memory. HyperMail was used as the archive and public forum for some of the key discussions that defined the emerging ideas of the early Web.

Acknowledgements: Many thanks to Amit Sheth who bring me in Contact with Tom