Extreme Markup Languages 2003

Montréal, Québec
August 4-8, 2003

A semantic integration methodology

Steven R. Newcomb
Coolheads Consulting

Abstract

The heart of the semantic integration problem is how to tell when two statements are about the same subject. In some circles, this is known as the “co-referencing problem”. It is problem familiar to those who sift and scrub intelligence gathered from diverse sources, and it is known to be hard. One of the reasons that it’s hard is that some statements define (or contribute to the definition of) the subjects they’re talking about.

A semantic integration methodology

Table of Contents

EXTREME MARKUP LANGUAGES 2003

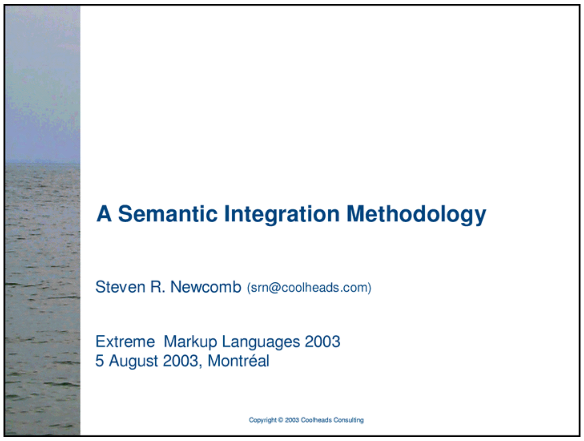
A semantic integration methodology

Steven R. Newcomb

This document is a set of annotated slides that was used by the author at Extreme Markup Languages 2003 to propose a definition of *semantic integration* and a Methodology for achieving it. The Methodology accommodates diverse worldviews, and it compromises neither the independence of knowledge contributors, nor the integrity of their contributions.

§ Introduction

Figure 1



A Semantic Integration Methodology

Steven R. Newcomb (srn@coolheads.com)

Extreme Markup Languages 2003
5 August 2003, Montréal

Copyright © 2003 Coolheads Consulting

The heart of the semantic integration problem is how to tell when two statements are about the same subject. In some circles, this is known as the *co-referencing problem*. It is a problem familiar to those who sift and scrub intelligence gathered from diverse sources, and it is known to be hard. One of the reasons that it's hard is that some statements define (or contribute to the definition of) the subjects they're talking about.

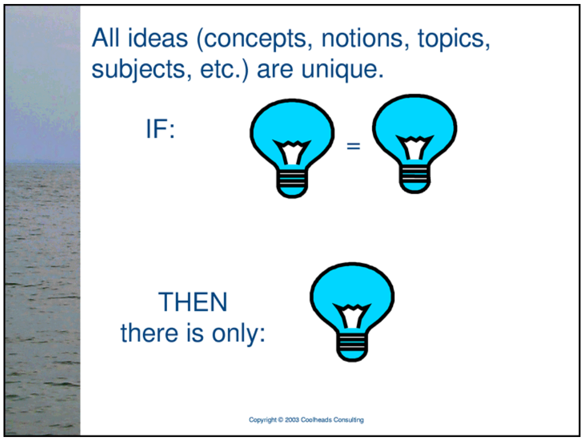
It is reasonable to assume that the publicly-available information on the Semantic Web will be *intended* to be semantically integrated by anyone with any other information. Semantic integration on the Semantic Web, then, could conceivably present a more tractable co-referencing problem than the one faced by the intelligence community, which gathers much information that was not intentionally contributed, and which was usually not created with the intent that it be semantically integrated with other information.

What form would be ideal for supplying arbitrary information to broad aggregations of knowledge, such as the Semantic Web, assuming that the supplier intends it to be most readily amenable to semantic integration with other information? How can such aggregations of knowledge be completely open with respect to semantics, including ontological semantics, and at the same time facilitate semantic integration in a meaningful way?

§ A problem statement

In these slides, I'm representing ideas — concepts, subjects of conversation — as lightbulbs. The lightbulbs represent "pure" subjects, quite apart from any symbols or other representations of them.

When we humans communicate with each other, we have to assume not only that we have some symbols (words, etc.) in common, but also that we share some common ideas. We necessarily assume that, at least some of the time, humans communicate so successfully and compellingly that they really do grasp the same idea. Normally, we also necessarily assume that the fact that two conscious entities are aware of an idea, or that they are talking about an idea, does not cause there to be two ideas. We also assume that all ideas are unique — that, in some sense, ideas have identity.



All ideas (concepts, notions, topics, subjects, etc.) are unique.

IF: =

THEN there is only:

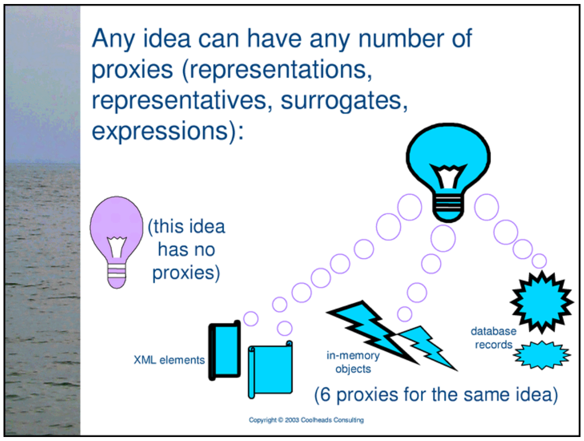
However, any idea can have any number of expressions; it can be the subject of any number of statements, in any number of conversations.

When we decide to manage information according to the ideas to which it is relevant, we can create indexes, such as the indexes often found in the backmatter of printed books. Each entry in an index is a proxy for a single subject of conversation, and the pages in the book that are considered relevant to that subject are, in some sense, properties of that proxy.

More generally, there are many occasions in which a specific unit of information serves as a kind of surrogate for a subject of conversation. In Topic Maps, for example, XML elements called `<topic>`s serve as proxies for subjects.

Subject proxies are not always pieces of text. For example, when a Topic Map tool has read an XML topic map document, it typically has a set of in-memory objects, each of which serves as a surrogate for a specific subject. For another example, it is often useful to regard certain kinds of records in relational databases as proxies for specific subjects.

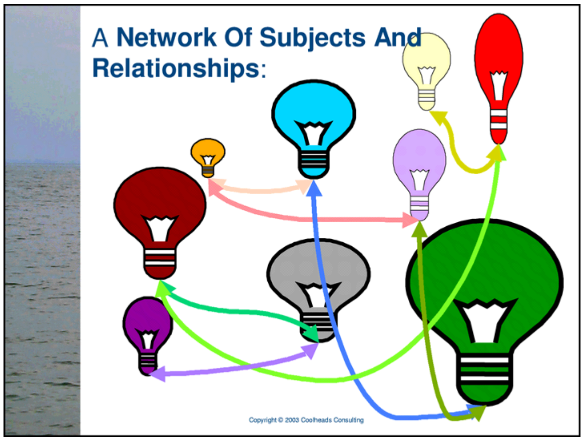
In this presentation, I’m using a visual vocabulary that provides three kinds of subject proxies: scrolls are XML elements used as subject proxies, lightning bolts are in-memory objects used as subject proxies, and 16-pointed stars are relational database records used as subject proxies. (There are many other kinds of subject proxies, of course.) N.B.: Lightbulbs are *not* subject proxies; they are the subjects themselves.



Any idea can have any number of proxies (representations, representatives, surrogates, expressions):

All human communication can be seen as expressions of relationships between subjects of conversation. Any piece of information, when understood, can be understood as a network consisting of subjects and the relationships between those subjects. If there are multiple interpretations of a given piece of information, each such interpretation is such a network.

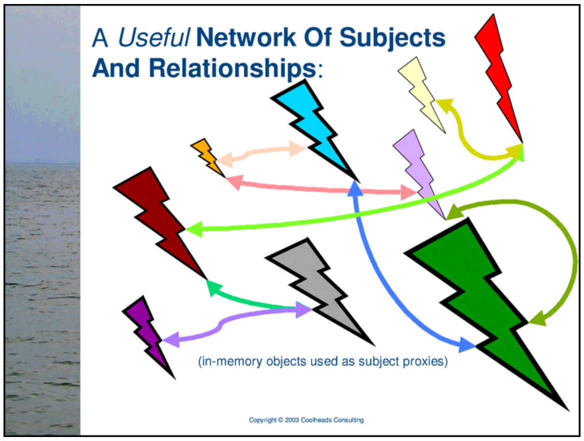
This slide depicts the world of ideas/subjects (lightbulbs). In this world are found the meanings of human expressions, rather than the expressions themselves. Or there may not be any expression whatsoever that corresponds to this constellation of ideas (this **Network of Subjects and Relationships**). This network may, for example, be someone’s unconscious, unexpressed worldview. Or it may represent some combination of views that have been conceived separately, but that nobody has actually combined. (The potentially enormous value to society of facilitating such combinations, and making them useful and visible, is the primary motivation for the development of the Semantic Integration Methodology described in this presentation.)



A **Network Of Subjects And Relationships**:

Copyright © 2003 Coolheads Consulting

The network of subjects and relationships shown in the previous slide is useless, because it is not being communicated or represented in the real world. In this slide, the same network of subjects and relationships has been made real by endowing each subject with a proxy. (In this particular slide, the proxies are all in-memory objects — lightning bolts.) Thus, the network of subjects and relationships is tangible, processable, and *useful*.

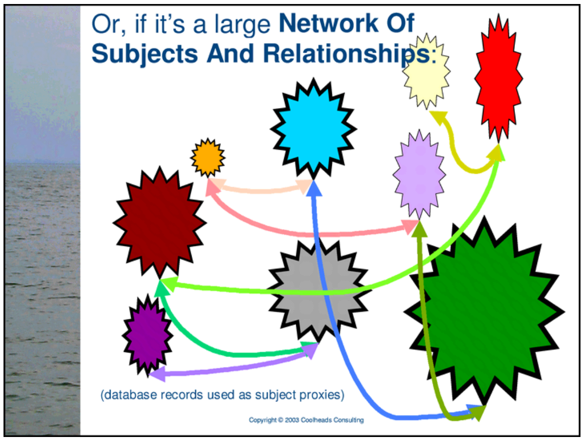


A *Useful* **Network Of Subjects And Relationships**:

(in-memory objects used as subject proxies)

Copyright © 2003 Coolheads Consulting

Computer memory is always a limited resource, but there the quantity of knowledge that can be usefully represented in a network of subjects and relationships is unbounded. In this slide, we’re using a relational database to increase our scale — to allow us to manage more subjects (more subject proxies) than we could handle using only in-memory objects.

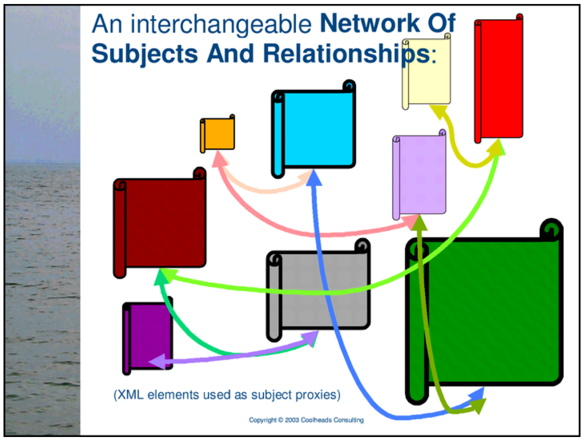


Or, if it's a large **Network Of Subjects And Relationships:**

(database records used as subject proxies)

Copyright © 2003 Coolheads Consulting

We can represent a network of subjects and relationships in XML, too, usually for the purpose of information interchange. Perhaps all the proxies in this slide are elements in a single XML document, or perhaps they are distributed across several XML documents.



An interchangeable **Network Of Subjects And Relationships:**

(XML elements used as subject proxies)

Copyright © 2003 Coolheads Consulting

It's important to understand that there is no network of subjects unless the subjects have relationships to one another.

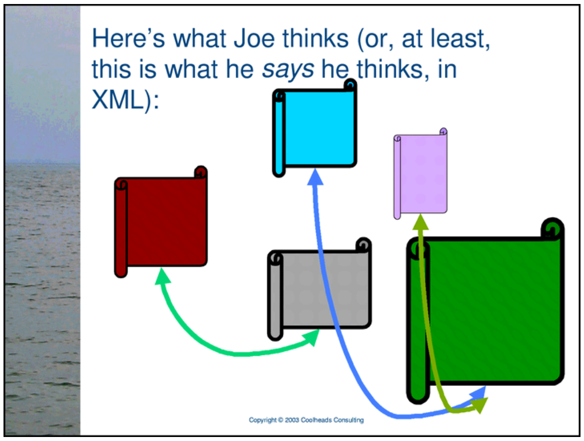
Without relationships, nothing is being said. All statements are statements of relationships among subjects.

Without relationships, nothing is being said.

All statements are statements of relationships among subjects.

Copyright © 2003 Coolheads Consulting

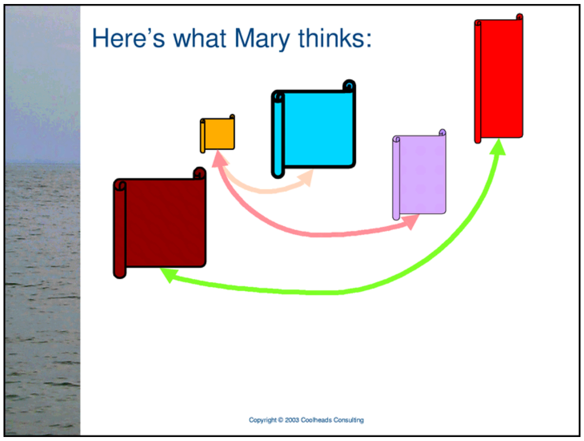
People can publish their views in the form of XML documents that represent networks of subjects and relationships. Here is Joe's XML document.



Here’s what Joe thinks (or, at least, this is what he *says* he thinks, in XML):

Copyright © 2003 Coolheads Consulting

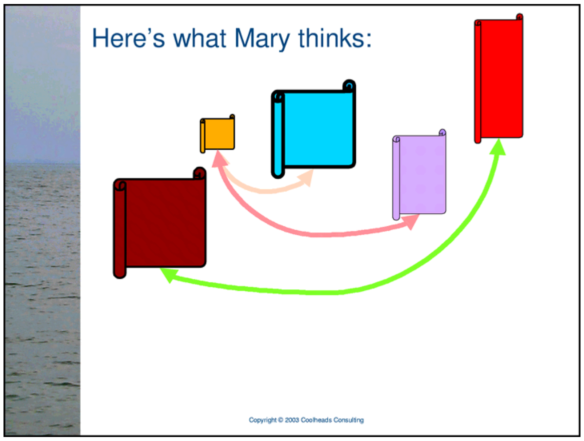
Here’s Mary’s XML network of subjects and relationships.



Here’s what Mary thinks:

Copyright © 2003 Coolheads Consulting

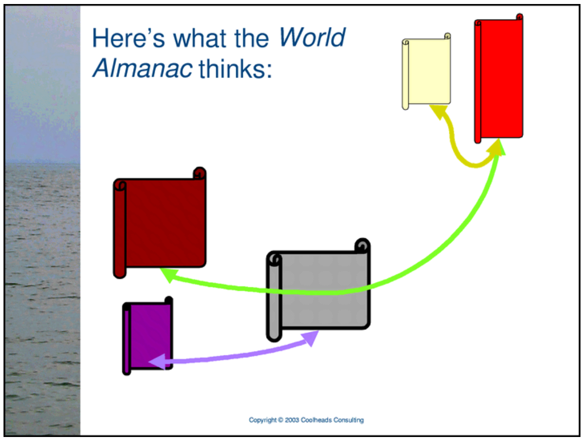
Here’s the *World Almanac’s* XML network of subjects and relationships.



Here’s what Mary thinks:

Copyright © 2003 Coolheads Consulting

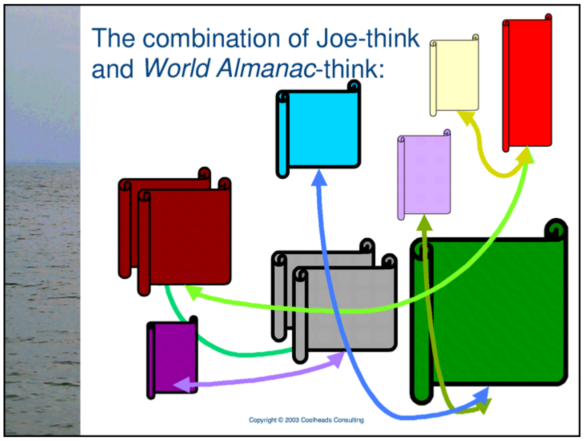
Joe and Mary say different things three of the same subjects. Wouldn’t it be great if, when we needed to know something about one of those subjects, we could know what both Joe and Mary thought?



Here's what the *World Almanac* thinks:

Copyright © 2003 Coolheads Consulting

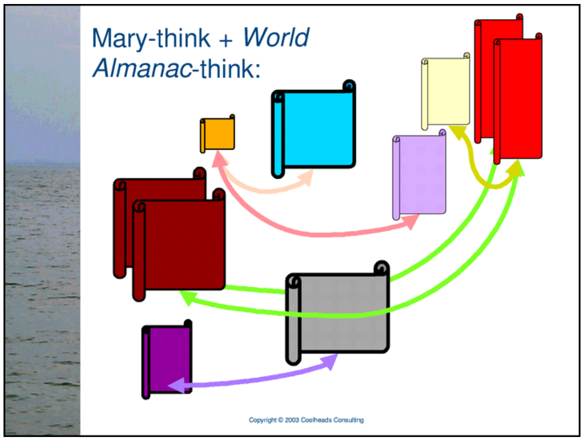
Joe and the *World Almanac* have different things to say about two subjects.



The combination of Joe-think and *World Almanac*-think:

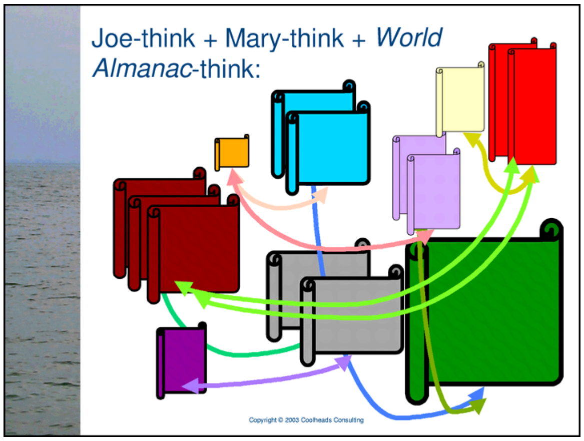
Copyright © 2003 Coolheads Consulting

In addition to the different things they have to say, Mary and the *World Almanac* make exactly the same statement, here depicted as a pair of light-green double-headed arrows. Each arrow has a proxy for the brown subject as one role-player in the relationship, and a proxy for the red subject as the other role-player.



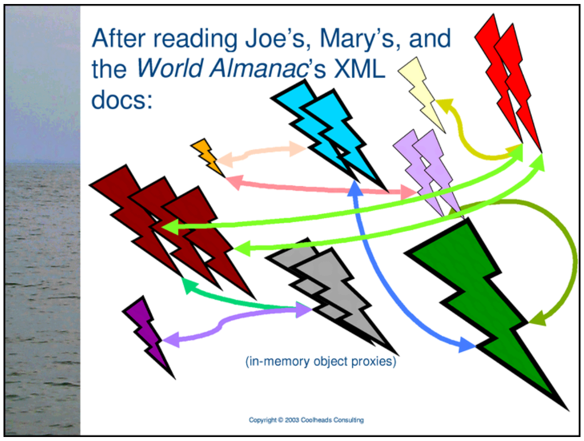
Mary-think + *World Almanac*-think:

Copyright © 2003 Coolheads Consulting



Joe-think + Mary-think + *World Almanac*-think:

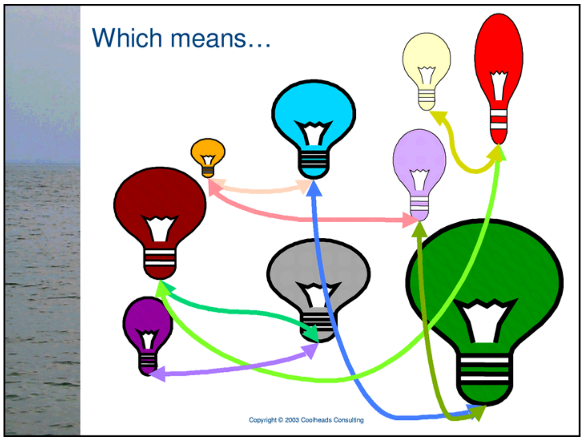
Let’s imagine that we have software — a Methodology engine — that can read Joe’s, Mary’s, and the *World Almanac’s* XML representations of their respective Networks of Subjects and Relationships, and translate them into corresponding sets of in-memory objects. Despite the fact that the result includes multiple in-memory proxies for some subjects...



After reading Joe’s, Mary’s, and the *World Almanac*’s XML docs:

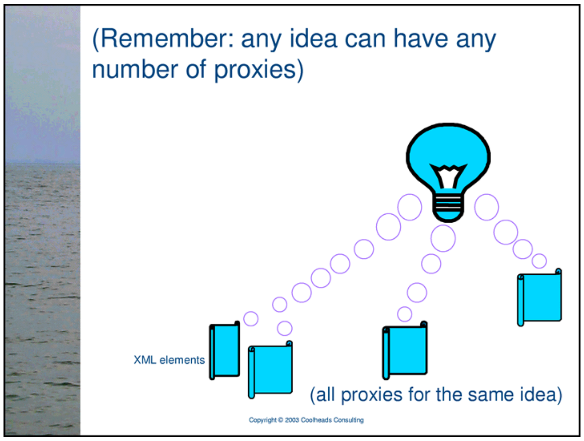
(in-memory object proxies)

… the combined network really ought to be the network of subjects and relationships depicted in this slide, because …



Which means…

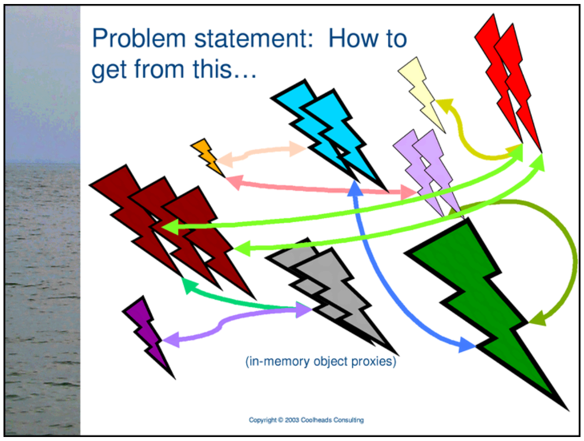
… any subject can have any number of proxies, but it is still a single, unique subject.



(Remember: any idea can have any number of proxies)

(all proxies for the same idea)

The problem that the Methodology addresses is the problem of combining multiple, independently-conceived representations of Networks of Subjects and Relationships, with their separate, partially redundant proxies for the same subjects, in such a way that …



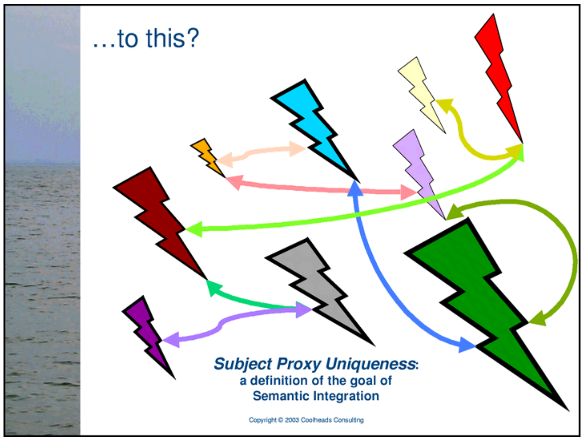
Problem statement: How to get from this…

(in-memory object proxies)

… for each subject, there is only one proxy, but no information has been lost. The Methodology's definition of *semantic integration* is *subject proxy uniqueness*.

The Methodology answers the questions:

- How can any subject be provided with one or more proxies or surrogates for itself? What design features should all such proxies have in common, and what features should be left open, in order to be sure that no subject will be excluded?
- How, exactly, can we distinguish whether two proxies have the same subject? In other words, how does a proxy identify its subject?

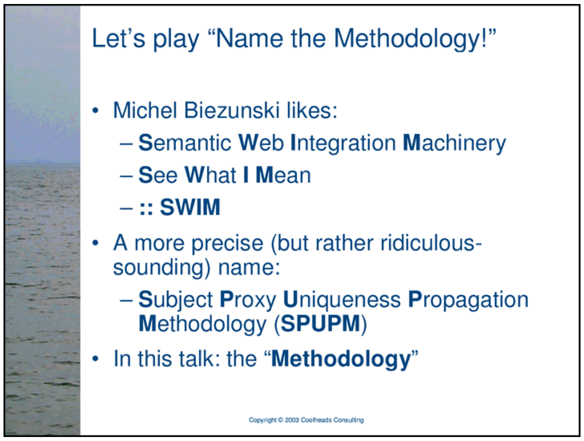


…to this?

***Subject Proxy Uniqueness*:**
a definition of the goal of Semantic Integration

Copyright © 2003 Coolheads Consulting

§ A solution statement

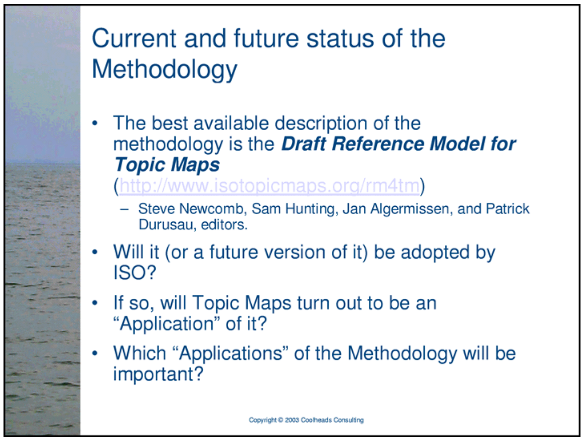


Let's play "Name the Methodology!"

- Michel Biezunski likes:
  - **S**emantic **W**eb **I**ntegration **M**achinery
  - **S**ee **W**hat **I** **M**ean
  - **:: SWIM**
- A more precise (but rather ridiculous-sounding) name:
  - **S**ubject **P**roxy **U**niqueness **P**ropagation **M**ethodology (**SPUPM**)
- In this talk: the "**Methodology**"

Copyright © 2003 Coolheads Consulting

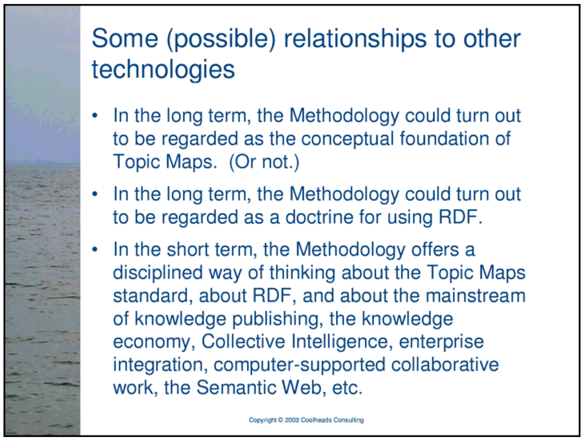
The Methodology is expressed as a potential ISO standard, as the *Draft Reference Model for Topic Maps* (**http://www.isotopicmaps.org/rm4tm**). The future of the Methodology is unknown.



Current and future status of the Methodology

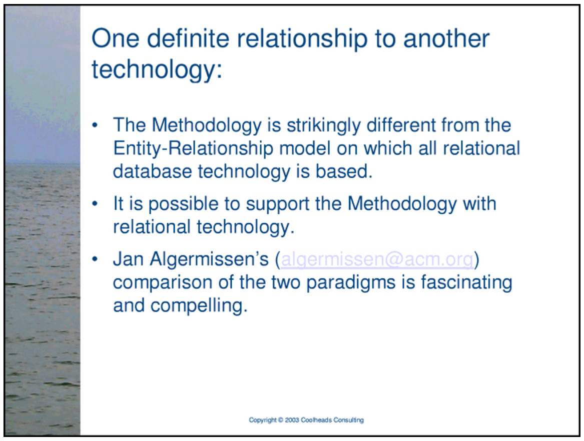
- The best available description of the methodology is the ***Draft Reference Model for Topic Maps*** (http://www.isotopicmaps.org/rm4tm)
  - Steve Newcomb, Sam Hunting, Jan Algermissen, and Patrick Durusau, editors.
- Will it (or a future version of it) be adopted by ISO?
- If so, will Topic Maps turn out to be an "Application" of it?
- Which "Applications" of the Methodology will be important?

Copyright © 2003 Coolheads Consulting



Some (possible) relationships to other technologies

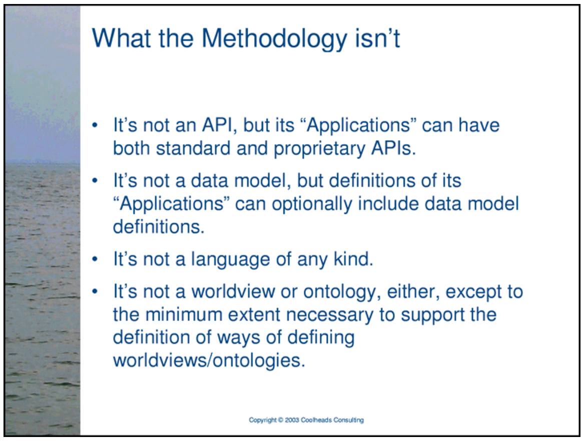
- In the long term, the Methodology could turn out to be regarded as the conceptual foundation of Topic Maps. (Or not.)
- In the long term, the Methodology could turn out to be regarded as a doctrine for using RDF.
- In the short term, the Methodology offers a disciplined way of thinking about the Topic Maps standard, about RDF, and about the mainstream of knowledge publishing, the knowledge economy, Collective Intelligence, enterprise integration, computer-supported collaborative work, the Semantic Web, etc.



One definite relationship to another technology:

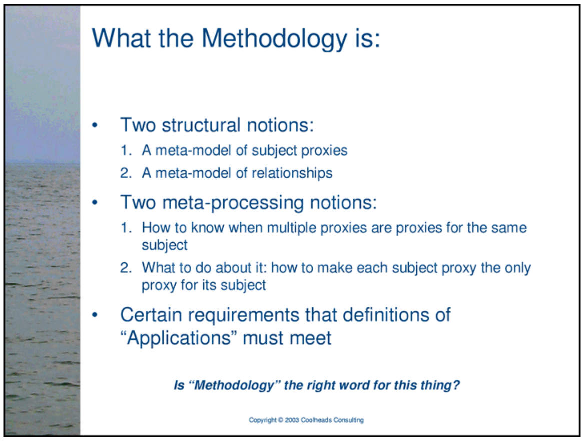
- The Methodology is strikingly different from the Entity-Relationship model on which all relational database technology is based.
- It is possible to support the Methodology with relational technology.
- Jan Algermissen's (algermissen@acm.org) comparison of the two paradigms is fascinating and compelling.

The word “Applications” has special meaning here, which is why it appears in doublequotes. “Applications” of the Methodology are not implementations of it; they are more like languages, with built-in notions about how one may determine whether two proxies are, in fact, proxies for the same subject.



What the Methodology isn't

- It's not an API, but its “Applications” can have both standard and proprietary APIs.
- It's not a data model, but definitions of its “Applications” can optionally include data model definitions.
- It's not a language of any kind.
- It's not a worldview or ontology, either, except to the minimum extent necessary to support the definition of ways of defining worldviews/ontologies.



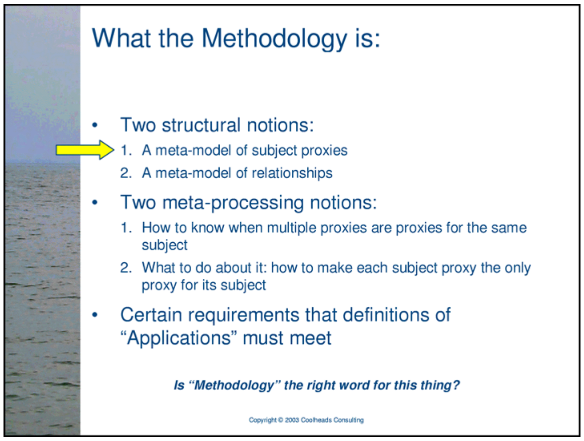
What the Methodology is:

- Two structural notions:
  1. A meta-model of subject proxies
  2. A meta-model of relationships
- Two meta-processing notions:
  1. How to know when multiple proxies are proxies for the same subject
  2. What to do about it: how to make each subject proxy the only proxy for its subject
- Certain requirements that definitions of "Applications" must meet

Is "Methodology" the right word for this thing?

Copyright © 2003 Coolheads Consulting

We'll discuss each of the features of the Methodology in turn, beginning with the answer to the question, "What is a subject proxy, really?"

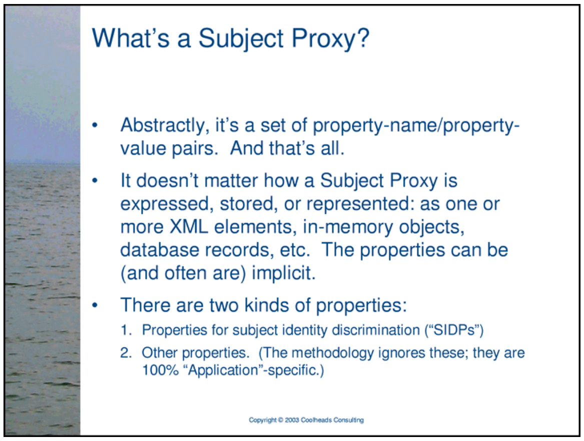


What the Methodology is:

- Two structural notions:
  1. A meta-model of subject proxies
  2. A meta-model of relationships
- Two meta-processing notions:
  1. How to know when multiple proxies are proxies for the same subject
  2. What to do about it: how to make each subject proxy the only proxy for its subject
- Certain requirements that definitions of "Applications" must meet

Is "Methodology" the right word for this thing?

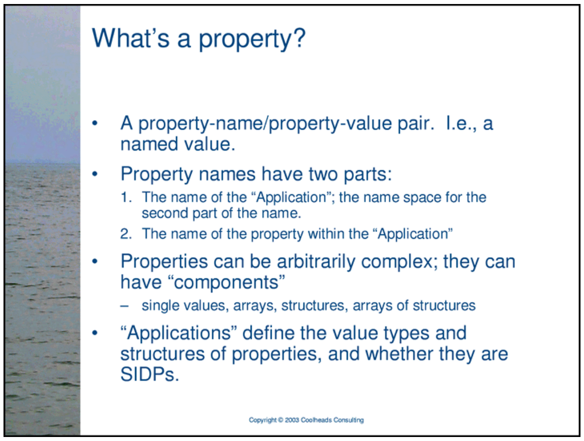
Copyright © 2003 Coolheads Consulting



What's a Subject Proxy?

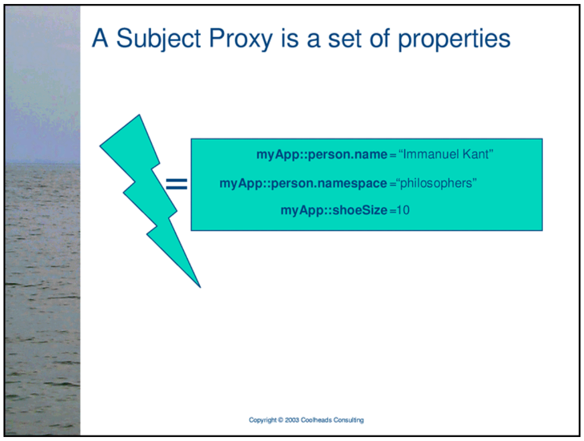
- Abstractly, it's a set of property-name/property-value pairs. And that's all.
- It doesn't matter how a Subject Proxy is expressed, stored, or represented: as one or more XML elements, in-memory objects, database records, etc. The properties can be (and often are) implicit.
- There are two kinds of properties:
  1. Properties for subject identity discrimination ("SIDPs")
  2. Other properties. (The methodology ignores these; they are 100% "Application"-specific.)

Copyright © 2003 Coolheads Consulting



What's a property?

- A property-name/property-value pair. I.e., a named value.
- Property names have two parts:
  1. The name of the "Application"; the name space for the second part of the name.
  2. The name of the property within the "Application"
- Properties can be arbitrarily complex; they can have "components"
  – single values, arrays, structures, arrays of structures
- "Applications" define the value types and structures of properties, and whether they are SIDPs.



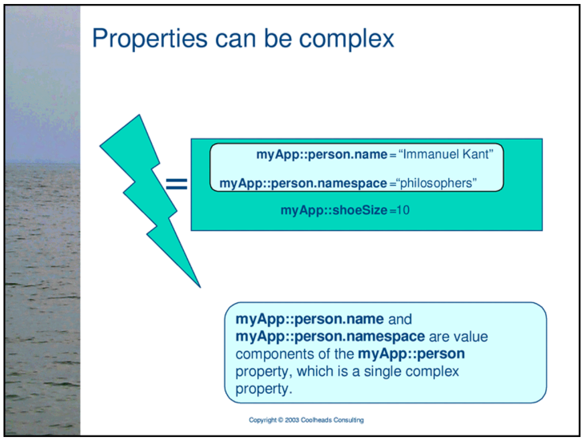
A Subject Proxy is a set of properties

=

myApp::person.name = "Immanuel Kant"

myApp::person.namespace ="philosophers"

myApp::shoeSize =10



Properties can be complex

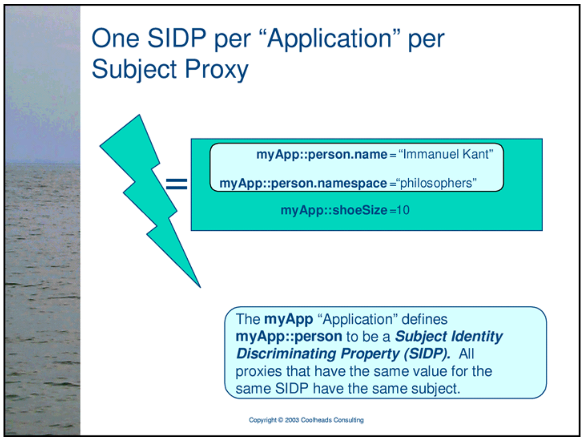
=

myApp::person.name = "Immanuel Kant"

myApp::person.namespace ="philosophers"

myApp::shoeSize =10

myApp::person.name and **myApp::person.namespace** are value components of the **myApp::person** property, which is a single complex property.



One SIDP per “Application” per Subject Proxy

=

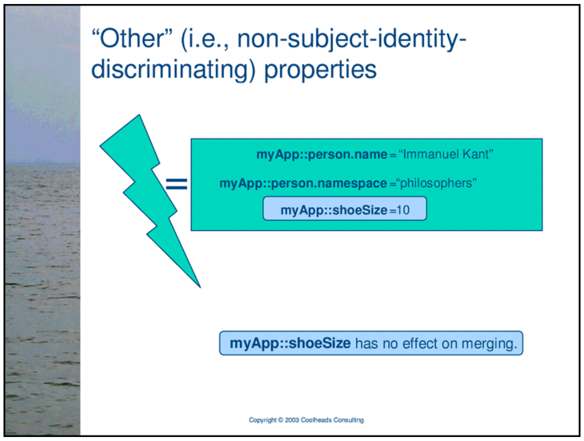
myApp::person.name = “Immanuel Kant”

myApp::person.namespace =“philosophers”

myApp::shoeSize =10

The **myApp** “Application” defines **myApp::person** to be a ***Subject Identity Discriminating Property (SIDP).*** All proxies that have the same value for the same SIDP have the same subject.

Copyright © 2003 Coolheads Consulting



“Other” (i.e., non-subject-identity-discriminating) properties

=

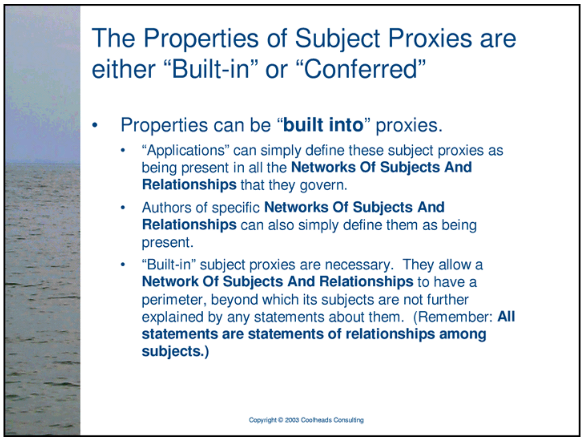
myApp::person.name = “Immanuel Kant”

myApp::person.namespace =“philosophers”

myApp::shoeSize =10

myApp::shoeSize has no effect on merging.

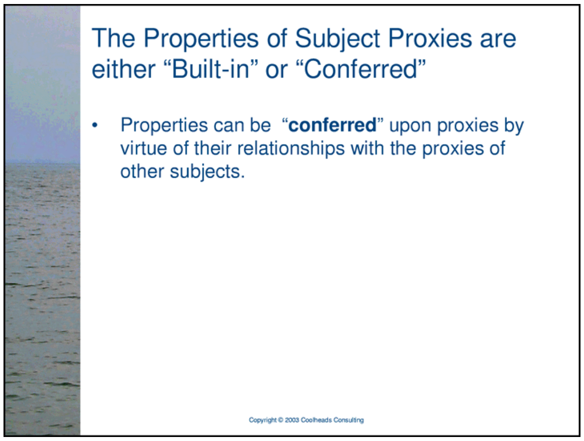
Copyright © 2003 Coolheads Consulting



The Properties of Subject Proxies are either “Built-in” or “Conferred”

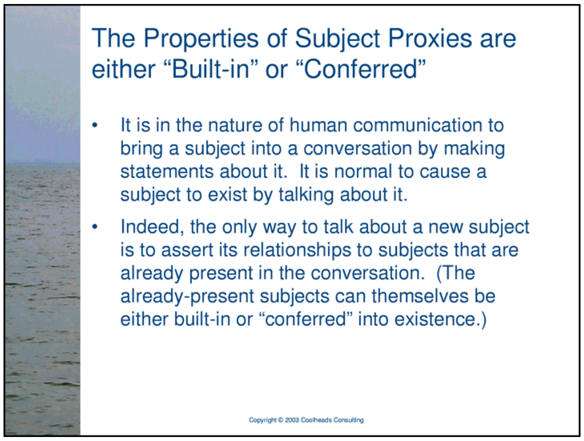
- Properties can be “**built into**” proxies.
  - “Applications” can simply define these subject proxies as being present in all the **Networks Of Subjects And Relationships** that they govern.
  - Authors of specific **Networks Of Subjects And Relationships** can also simply define them as being present.
  - “Built-in” subject proxies are necessary. They allow a **Network Of Subjects And Relationships** to have a perimeter, beyond which its subjects are not further explained by any statements about them. (Remember: **All statements are statements of relationships among subjects.)**

Copyright © 2003 Coolheads Consulting



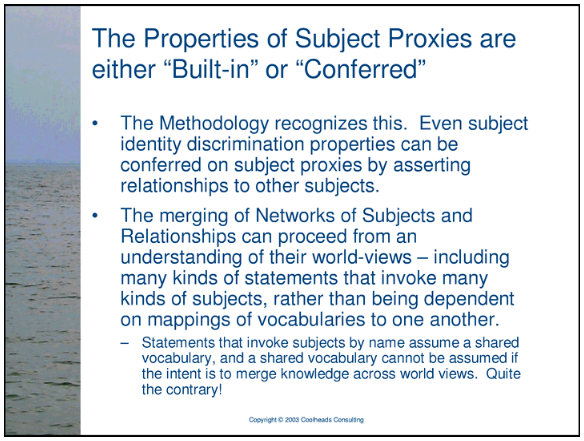
The Properties of Subject Proxies are either “Built-in” or “Conferred”

- Properties can be “**conferred**” upon proxies by virtue of their relationships with the proxies of other subjects.



The Properties of Subject Proxies are either “Built-in” or “Conferred”

- It is in the nature of human communication to bring a subject into a conversation by making statements about it. It is normal to cause a subject to exist by talking about it.
- Indeed, the only way to talk about a new subject is to assert its relationships to subjects that are already present in the conversation. (The already-present subjects can themselves be either built-in or “conferred” into existence.)



The Properties of Subject Proxies are either “Built-in” or “Conferred”

- The Methodology recognizes this. Even subject identity discrimination properties can be conferred on subject proxies by asserting relationships to other subjects.
- The merging of Networks of Subjects and Relationships can proceed from an understanding of their world-views – including many kinds of statements that invoke many kinds of subjects, rather than being dependent on mappings of vocabularies to one another.
  - Statements that invoke subjects by name assume a shared vocabulary, and a shared vocabulary cannot be assumed if the intent is to merge knowledge across world views. Quite the contrary!

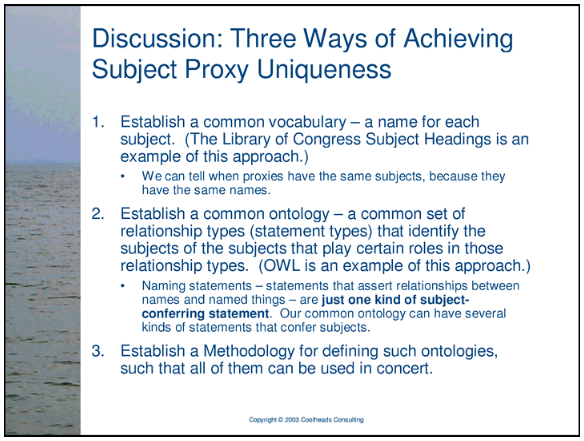
Establishing authorities for the names of subjects is something that many (and, arguably, most) people do, and that’s the problem. At the scale of the Semantic Web, the idea that a few subject naming authorities will give names to everything we need to talk about will not work very well as the basis of semantic integration. There are many reasons why it won’t work. One of them is that, as some wag once noted, the greatest thing about standards is that there are so many to choose from. But the basic problem is not that there are so many “standard” vocabularies, etc.; the basic problem is human nature, and in the nature of human communication. It can be argued that it is impossible to write pointed prose without either using existing terms idiosyncratically, or inventing new terms. This may be the true nature of the Curse of Babel — that human communication necessarily invents itself, to at least some degree, whenever it occurs. Again: a statement can be about a subject, and it may also define (or contribute to the definition of) the thing it’s talking about.

So, if common terminology — common names for subjects — is not a scalable basis on which the Semantic Web can merge proxies for identical subjects, on what basis can it be done?

Well, it can be done on the basis of a common ontology, if the common ontology provides a logical basis on which statements about subjects can identify them, for all purposes of deciding whether any two subject proxies are proxies for the same or different subjects. Even an ontology with very few, very simple types of assertions can provide such a basis, if they are all widely understood and honored. (Indeed, this is the approach used in the SAM [Standard Application Model] of Topic Maps, in which subjects can be asserted to have “subject indicators”. The “subject indicator” approach is more practical than attempting to get everyone to use the same subject identifiers. The subject indicators — arbitrary pieces of subject-describing information — are, in effect, the subject identifiers, but they have the added virtue of actually describing the subject in some compelling way. Subject indicators at least provide a more compelling basis for recognizing subject identity than, say, most URLs would normally be. Furthermore, the *context* of a subject indicator — the location in which it appears — can make it far more compelling and authoritative, and far more descriptive of its subject, than any context-independent name could possibly be.)

Unfortunately, the development of a single, universal ontology for subject identification — an ontology that everyone in the world will use to identify all subjects, forever — is a quixotic endeavor. We in the SGML/XML world know this in our very bones, because it is so similar to such ill-starred, scopeless ideas as the One True “ODA [Office Document Architecture]”. Any attempt to realize such an idea will probably absorb whatever resources are allocated to it, but without yielding the intended result.

No, in order to achieve the goal of Web-scale semantic integration without compromising the semantic authority of each information contributor, we have to take another step back. We have to content ourselves with saying how ontologies for subject identification and discrimination must be defined, and to provide a basis for diverse ontology definitions such that the semantic integration of diverse information expressed in terms of those ontologies is facilitated. The Methodology provides these things.

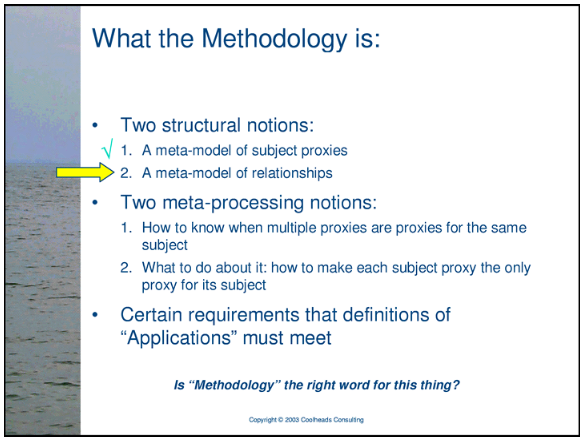


Discussion: Three Ways of Achieving Subject Proxy Uniqueness

1. Establish a common vocabulary – a name for each subject. (The Library of Congress Subject Headings is an example of this approach.)
   - We can tell when proxies have the same subjects, because they have the same names.
2. Establish a common ontology – a common set of relationship types (statement types) that identify the subjects of the subjects that play certain roles in those relationship types. (OWL is an example of this approach.)
   - Naming statements – statements that assert relationships between names and named things – are **just one kind of subject-conferring statement**. Our common ontology can have several kinds of statements that confer subjects.
3. Establish a Methodology for defining such ontologies, such that all of them can be used in concert.

Copyright © 2003 Coolheads Consulting

The next feature we’ll discuss is the Methodology’s answer to the question, “What’s a relationship?”

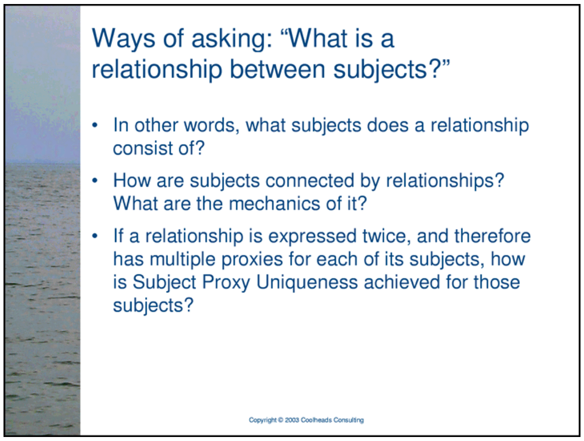


What the Methodology is:

- Two structural notions:
  1. √ A meta-model of subject proxies
  2. A meta-model of relationships
- Two meta-processing notions:
  1. How to know when multiple proxies are proxies for the same subject
  2. What to do about it: how to make each subject proxy the only proxy for its subject
- Certain requirements that definitions of “Applications” must meet

Is “Methodology” the right word for this thing?

Copyright © 2003 Coolheads Consulting

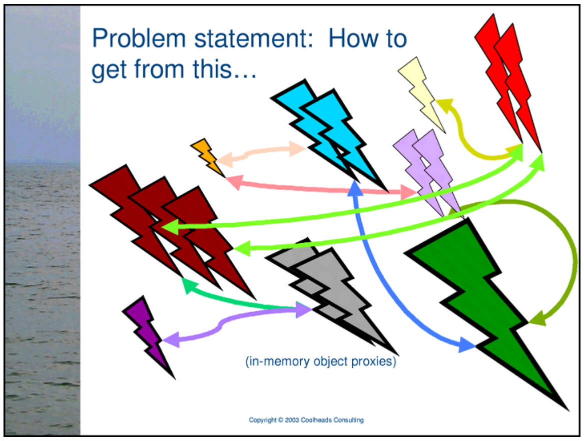


Ways of asking: “What is a relationship between subjects?”

- In other words, what subjects does a relationship consist of?
- How are subjects connected by relationships? What are the mechanics of it?
- If a relationship is expressed twice, and therefore has multiple proxies for each of its subjects, how is Subject Proxy Uniqueness achieved for those subjects?

Copyright © 2003 Coolheads Consulting

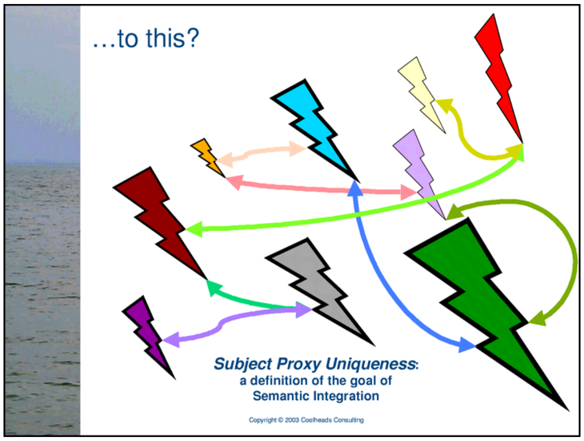
Note that we have two statements (here depicted as the two green double-headed arrows) that say the same thing. Since the statements are themselves subjects — *i.e.*, they are things that someone might want to talk about someday — they, too, must have subject proxies. If statements/relationships are represented as proxies, then …



Problem statement: How to get from this…

Copyright © 2003 Coolheads Consulting

… we need to know how to make them unique, too.

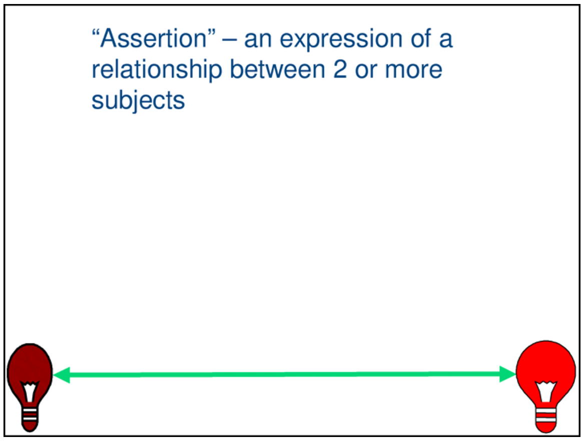


…to this?

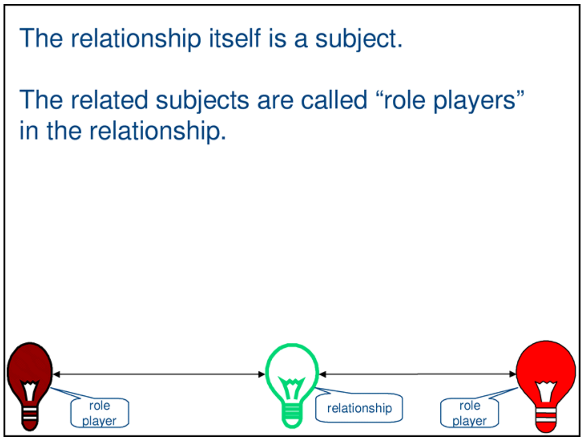
Subject Proxy Uniqueness:
a definition of the goal of
Semantic Integration

Copyright © 2003 Coolheads Consulting

In the Methodology, an expression of a relationship is called an *assertion*.

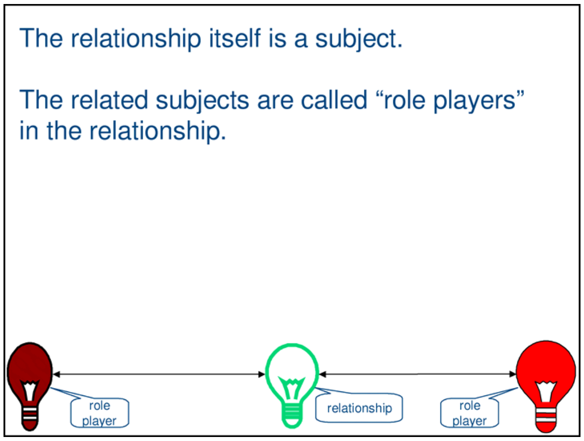


"Assertion" – an expression of a relationship between 2 or more subjects



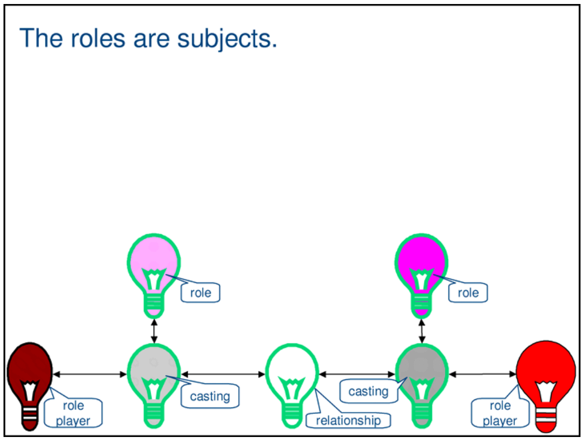
The relationship itself is a subject.

The related subjects are called "role players" in the relationship.

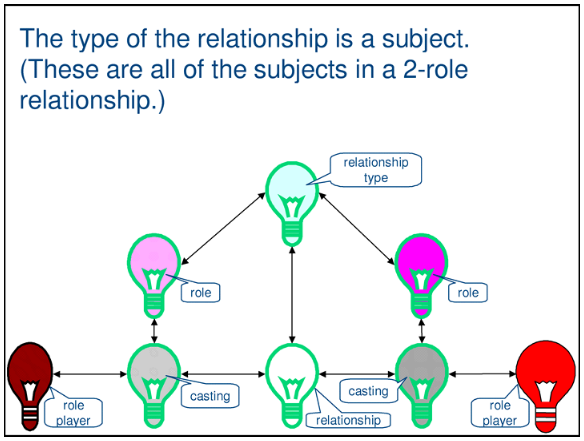


The relationship itself is a subject.

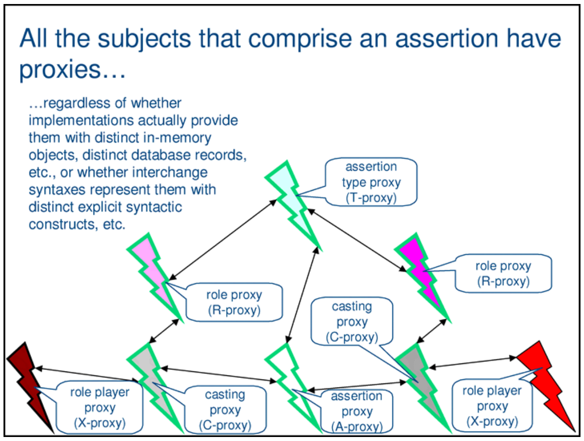
The related subjects are called “role players” in the relationship.



The roles are subjects.



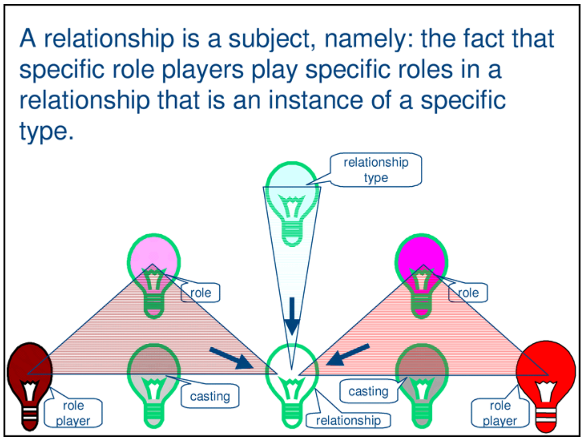
The type of the relationship is a subject. (These are all of the subjects in a 2-role relationship.)



All the subjects that comprise an assertion have proxies…

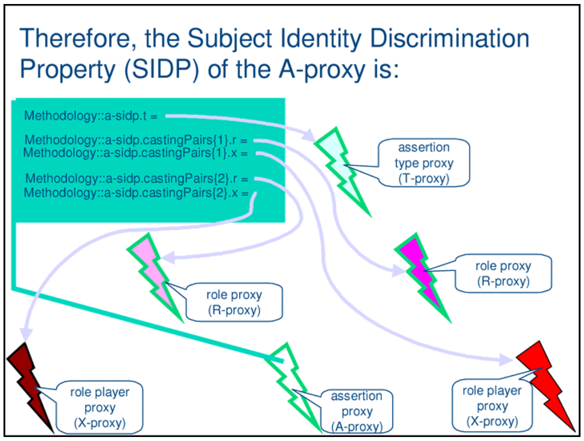
…regardless of whether implementations actually provide them with distinct in-memory objects, distinct database records, etc., or whether interchange syntaxes represent them with distinct explicit syntactic constructs, etc.

assertion type proxy (T-proxy)
role proxy (R-proxy)
role proxy (R-proxy)
casting proxy (C-proxy)
role player proxy (X-proxy)
casting proxy (C-proxy)
assertion proxy (A-proxy)
role player proxy (X-proxy)



A relationship is a subject, namely: the fact that specific role players play specific roles in a relationship that is an instance of a specific type.

The green box shows a single complex property — the SIDP [Subject Identity Discrimination Property] of the a-proxy at the bottom center. The property has five components, the value of each of which is a proxy. The top component, **a-sidp.t**, is the type of the assertion. The rest are the role/role-player pairs of the assertion. Together, these components uniquely identify the subject of the assertion.



Therefore, the Subject Identity Discrimination Property (SIDP) of the A-proxy is:

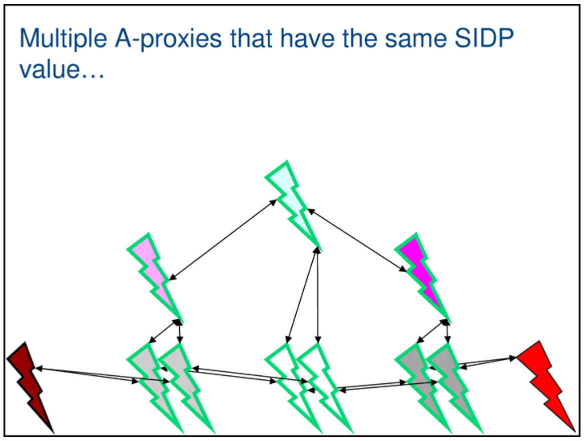
Methodology::a-sidp.t =

Methodology::a-sidp.castingPairs{1}.r =
Methodology::a-sidp.castingPairs{1}.x =

Methodology::a-sidp.castingPairs{2}.r =
Methodology::a-sidp.castingPairs{2}.x =

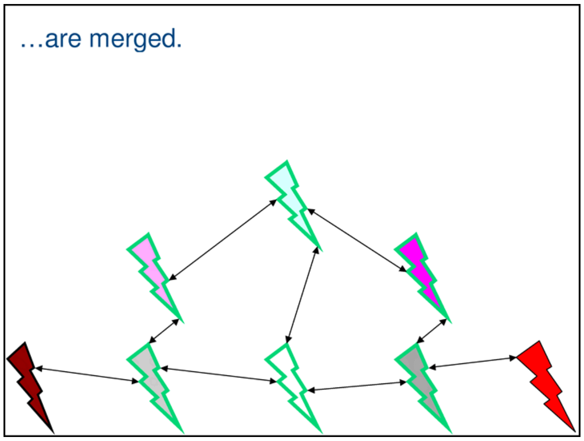
assertion type proxy (T-proxy)
role proxy (R-proxy)
role proxy (R-proxy)
role player proxy (X-proxy)
assertion proxy (A-proxy)
role player proxy (X-proxy)

If two assertions have the same subject (*i.e.*, they have equivalent SIDP values) …



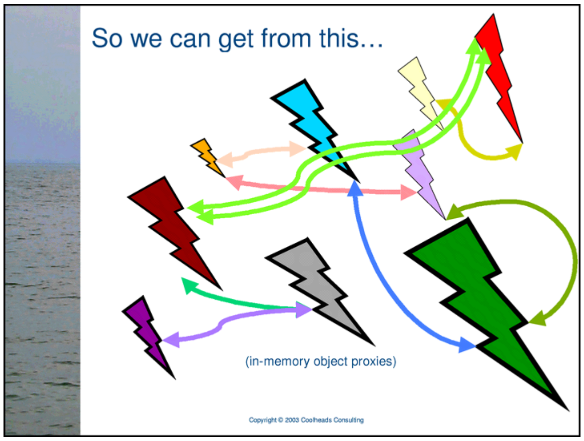
Multiple A-proxies that have the same SIDP value…

… they are merged.



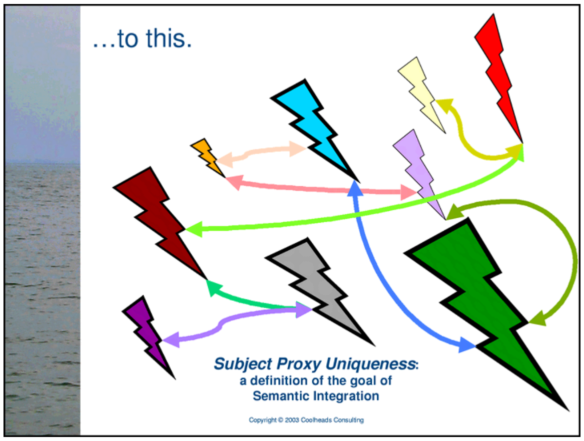
…are merged.

Since we can tell when two assertions are in fact saying exactly the same thing, …



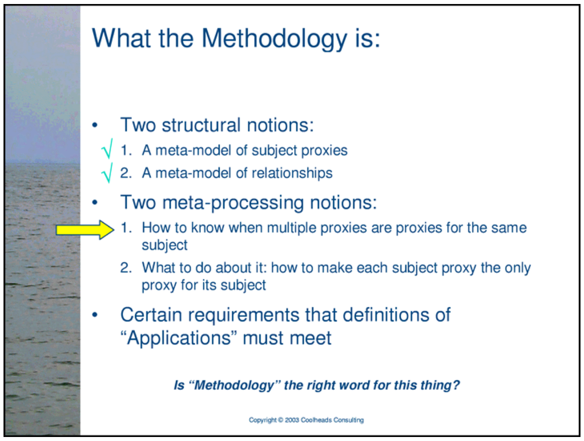
So we can get from this…

… we can eliminate the redundancy.



…to this.

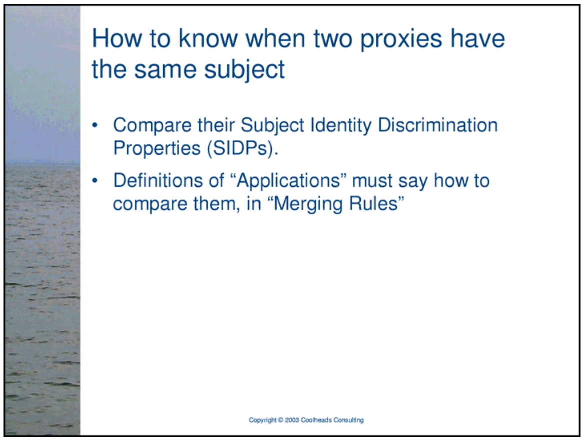
***Subject Proxy Uniqueness*:**
a definition of the goal of
Semantic Integration



What the Methodology is:

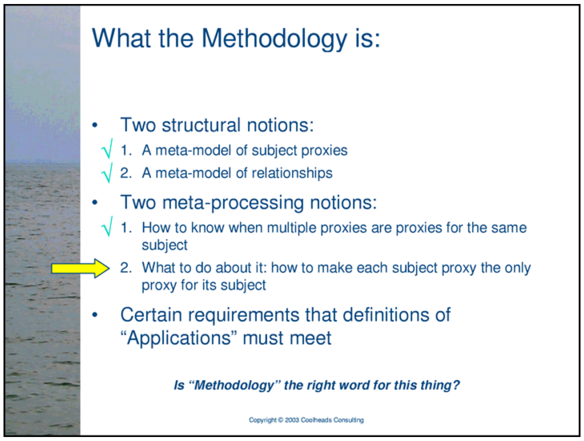
- Two structural notions:
  - √ 1. A meta-model of subject proxies
  - √ 2. A meta-model of relationships
- Two meta-processing notions:
  - ➡ 1. How to know when multiple proxies are proxies for the same subject
  - 2. What to do about it: how to make each subject proxy the only proxy for its subject
- Certain requirements that definitions of "Applications" must meet

Is "Methodology" the right word for this thing?



How to know when two proxies have the same subject

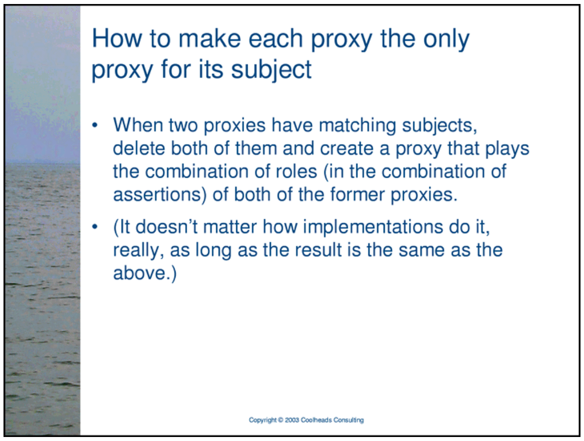
- Compare their Subject Identity Discrimination Properties (SIDPs).
- Definitions of "Applications" must say how to compare them, in "Merging Rules"



What the Methodology is:

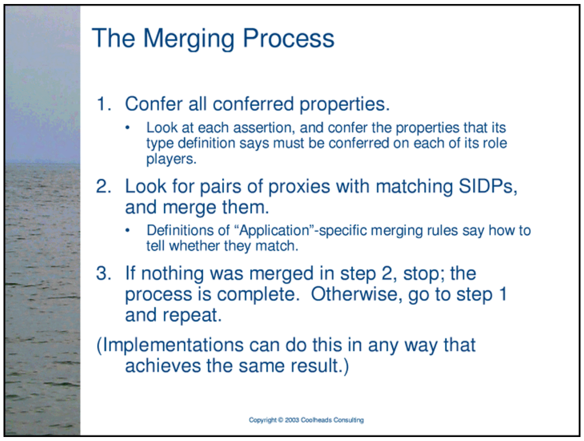
- Two structural notions:
  - √ 1. A meta-model of subject proxies
  - √ 2. A meta-model of relationships
- Two meta-processing notions:
  - √ 1. How to know when multiple proxies are proxies for the same subject
  - ➡ 2. What to do about it: how to make each subject proxy the only proxy for its subject
- Certain requirements that definitions of “Applications” must meet

Is “Methodology” the right word for this thing?



How to make each proxy the only proxy for its subject

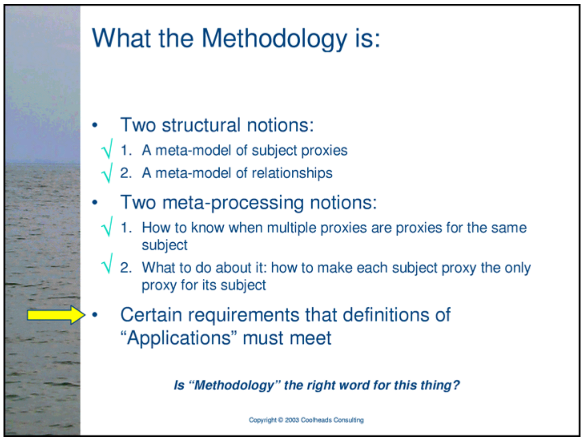
- When two proxies have matching subjects, delete both of them and create a proxy that plays the combination of roles (in the combination of assertions) of both of the former proxies.
- (It doesn’t matter how implementations do it, really, as long as the result is the same as the above.)



The Merging Process

1. Confer all conferred properties.
   - Look at each assertion, and confer the properties that its type definition says must be conferred on each of its role players.
2. Look for pairs of proxies with matching SIDPs, and merge them.
   - Definitions of “Application”-specific merging rules say how to tell whether they match.
3. If nothing was merged in step 2, stop; the process is complete. Otherwise, go to step 1 and repeat.

(Implementations can do this in any way that achieves the same result.)



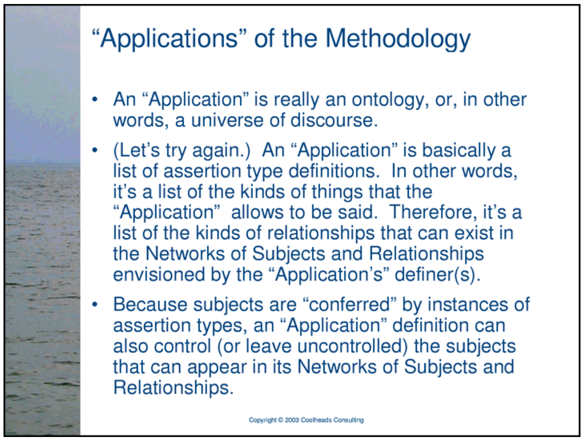
What the Methodology is:

- Two structural notions:
  - √ 1. A meta-model of subject proxies
  - √ 2. A meta-model of relationships
- Two meta-processing notions:
  - √ 1. How to know when multiple proxies are proxies for the same subject
  - √ 2. What to do about it: how to make each subject proxy the only proxy for its subject
- Certain requirements that definitions of "Applications" must meet

Is "Methodology" the right word for this thing?

Copyright © 2003 Coolheads Consulting

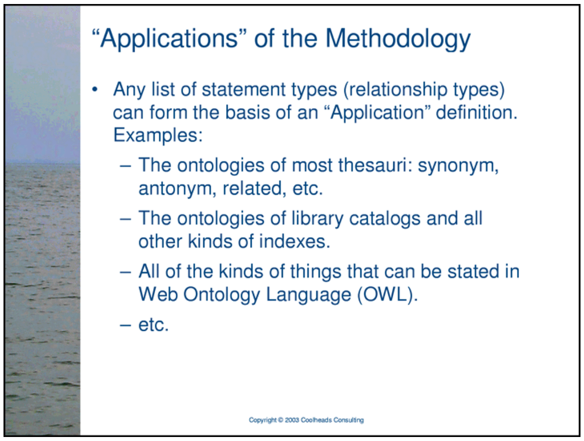
At its heart, the Methodology is a workable set of requirements for defining ontologies — "Applications" — that are intended to facilitate semantic integration. The Methodology demonstrates that it's possible to codify the requirements, even at its very high level of abstraction. The requirements themselves turn out to be roughly equal in complexity to the requirements for defining a document schema: it's not as simple as we might want it to be, but the complexity of the task is exactly as manageable as we decide to make it. In other words, the complexity of defining an ontology that supports semantic integration is proportional to the complexity of the integration that we're trying to accomplish with it.



"Applications" of the Methodology

- An "Application" is really an ontology, or, in other words, a universe of discourse.
- (Let's try again.) An "Application" is basically a list of assertion type definitions. In other words, it's a list of the kinds of things that the "Application" allows to be said. Therefore, it's a list of the kinds of relationships that can exist in the Networks of Subjects and Relationships envisioned by the "Application's" definer(s).
- Because subjects are "conferred" by instances of assertion types, an "Application" definition can also control (or leave uncontrolled) the subjects that can appear in its Networks of Subjects and Relationships.

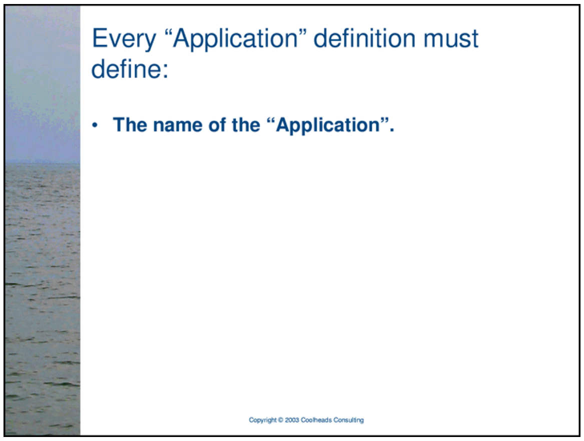
Copyright © 2003 Coolheads Consulting



"Applications" of the Methodology

- Any list of statement types (relationship types) can form the basis of an "Application" definition. Examples:
  - – The ontologies of most thesauri: synonym, antonym, related, etc.
  - – The ontologies of library catalogs and all other kinds of indexes.
  - – All of the kinds of things that can be stated in Web Ontology Language (OWL).
  - – etc.

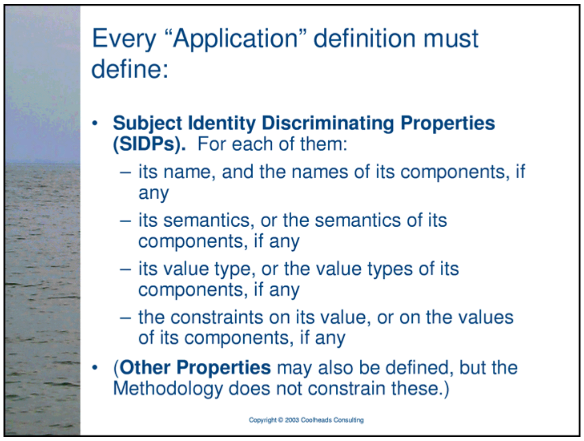
Copyright © 2003 Coolheads Consulting



Every “Application” definition must define:

- **The name of the “Application”.**

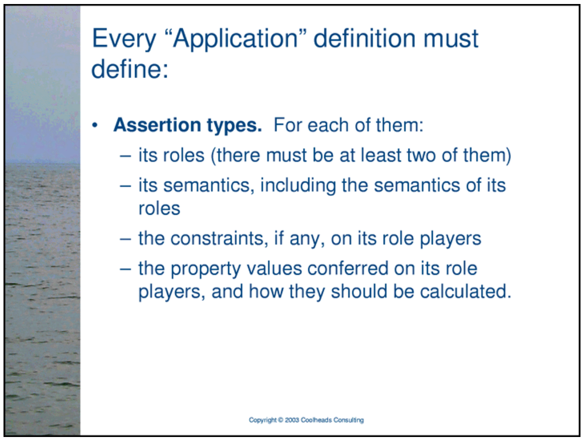
Copyright © 2003 Coolheads Consulting



Every “Application” definition must define:

- **Subject Identity Discriminating Properties (SIDPs).** For each of them:
  - its name, and the names of its components, if any
  - its semantics, or the semantics of its components, if any
  - its value type, or the value types of its components, if any
  - the constraints on its value, or on the values of its components, if any
- (**Other Properties** may also be defined, but the Methodology does not constrain these.)

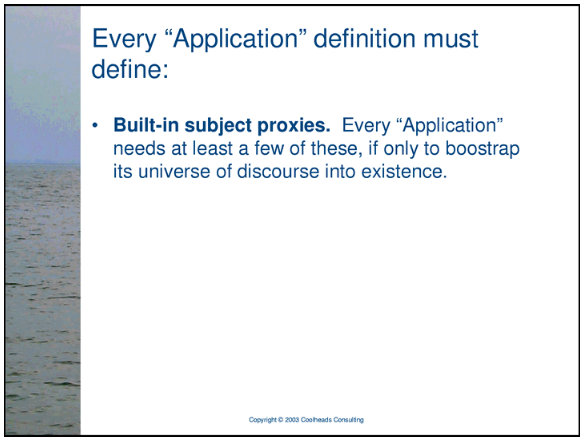
Copyright © 2003 Coolheads Consulting



Every “Application” definition must define:

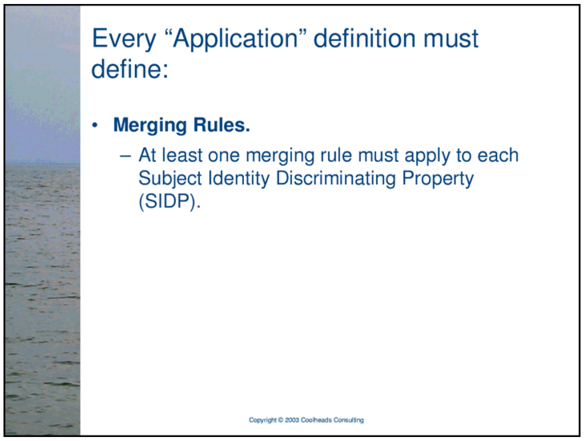
- **Assertion types.** For each of them:
  - its roles (there must be at least two of them)
  - its semantics, including the semantics of its roles
  - the constraints, if any, on its role players
  - the property values conferred on its role players, and how they should be calculated.

Copyright © 2003 Coolheads Consulting



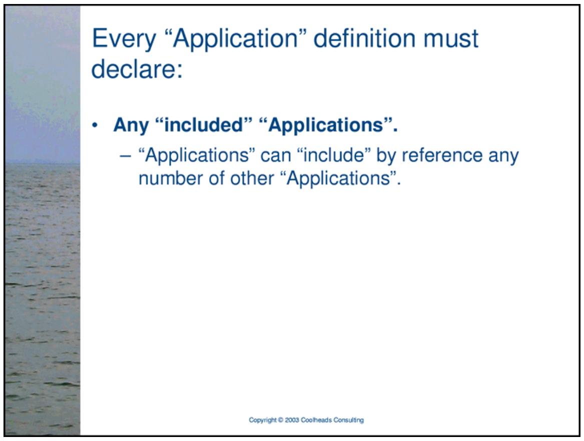
Every “Application” definition must define:

- **Built-in subject proxies.** Every “Application” needs at least a few of these, if only to boostrap its universe of discourse into existence.



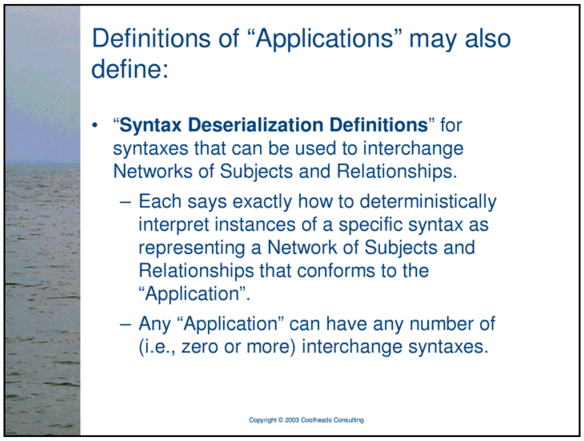
Every “Application” definition must define:

- **Merging Rules.**
  - At least one merging rule must apply to each Subject Identity Discriminating Property (SIDP).



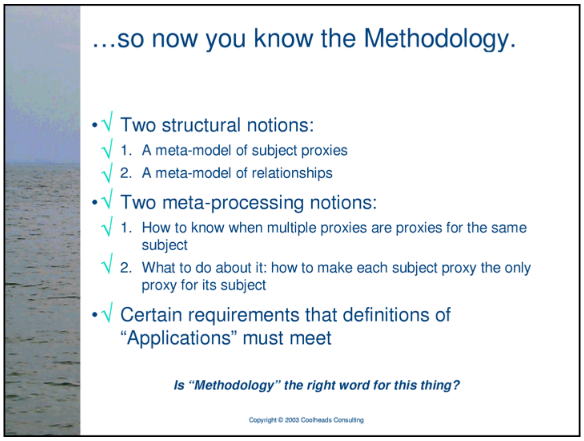
Every “Application” definition must declare:

- **Any “included” “Applications”.**
  - “Applications” can “include” by reference any number of other “Applications”.



Definitions of “Applications” may also define:

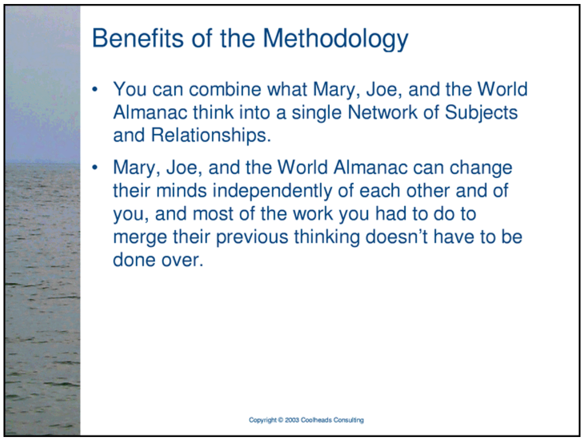
- “**Syntax Deserialization Definitions**” for syntaxes that can be used to interchange Networks of Subjects and Relationships.
  - Each says exactly how to deterministically interpret instances of a specific syntax as representing a Network of Subjects and Relationships that conforms to the “Application”.
  - Any “Application” can have any number of (i.e., zero or more) interchange syntaxes.



…so now you know the Methodology.

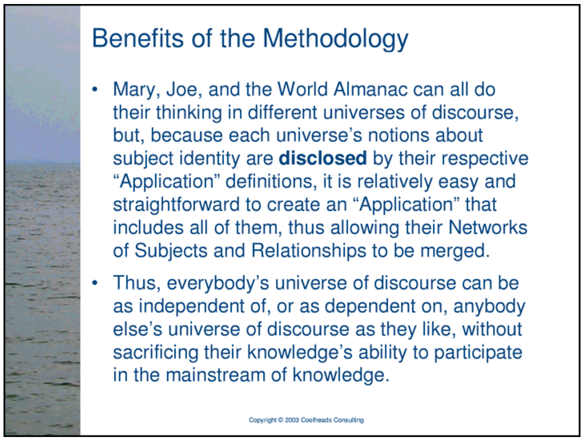
- √ Two structural notions:
  - √ 1. A meta-model of subject proxies
  - √ 2. A meta-model of relationships
- √ Two meta-processing notions:
  - √ 1. How to know when multiple proxies are proxies for the same subject
  - √ 2. What to do about it: how to make each subject proxy the only proxy for its subject
- √ Certain requirements that definitions of “Applications” must meet

Is “Methodology” the right word for this thing?



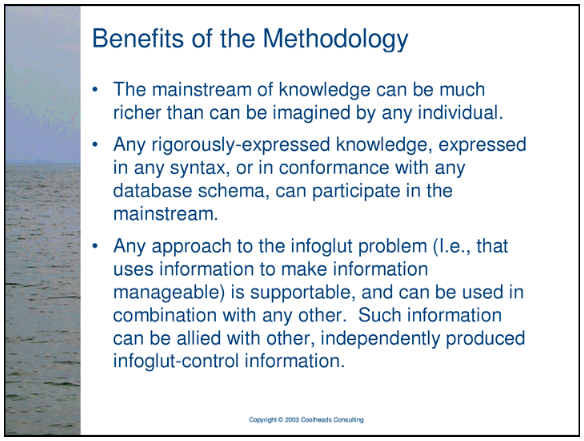
Benefits of the Methodology

- You can combine what Mary, Joe, and the World Almanac think into a single Network of Subjects and Relationships.
- Mary, Joe, and the World Almanac can change their minds independently of each other and of you, and most of the work you had to do to merge their previous thinking doesn’t have to be done over.



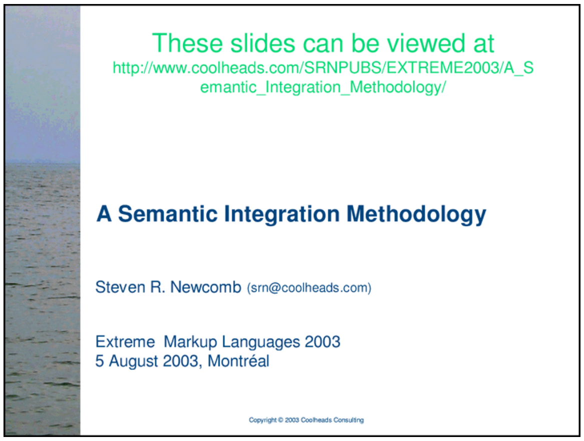
Benefits of the Methodology

- Mary, Joe, and the World Almanac can all do their thinking in different universes of discourse, but, because each universe's notions about subject identity are **disclosed** by their respective "Application" definitions, it is relatively easy and straightforward to create an "Application" that includes all of them, thus allowing their Networks of Subjects and Relationships to be merged.
- Thus, everybody's universe of discourse can be as independent of, or as dependent on, anybody else's universe of discourse as they like, without sacrificing their knowledge's ability to participate in the mainstream of knowledge.



Benefits of the Methodology

- The mainstream of knowledge can be much richer than can be imagined by any individual.
- Any rigorously-expressed knowledge, expressed in any syntax, or in conformance with any database schema, can participate in the mainstream.
- Any approach to the infoglut problem (I.e., that uses information to make information manageable) is supportable, and can be used in combination with any other. Such information can be allied with other, independently produced infoglut-control information.



These slides can be viewed at

http://www.coolheads.com/SRNPUBS/EXTREME2003/A_Semantic_Integration_Methodology/

A Semantic Integration Methodology

Steven R. Newcomb (srn@coolheads.com)

Extreme Markup Languages 2003
5 August 2003, Montréal

The Author

Steven R. Newcomb
Coolheads Consulting
208 Highview Drive
Blacksburg
VA
srn@coolheads.com

Steven R. Newcomb is an information architecture methodology pioneer, consultant, entrepreneur, and (former) university professor. He drafted and edited the ISO/IEC 13250:2000 and :2003 Topic Maps International Standard, also known as XTM [XML Topic Maps], and he drafted and edits the Reference Model for Topic Maps (2003). He served as editor of the ISO HyTime [Hypermedia/Time-based Structuring Language], ISO/IEC 10744:1992 and :1997), and of the ISO Standard Music Description Language (ISO/IEC 10743:1996). He founded and co-chairs the “Extreme Markup Languages” conference series of IDEAlliance, now in its ninth year.

Extreme Markup Languages 2003
Montréal, Québec, August 4-8, 2003
This paper was formatted from XML source via XSL
by Mulberry Technologies, Inc.