+Part 1: Knowledge Driven Architecture – Short Extracts

Part 2: Transitioning to Semantic Cloud
Part 3: After SOA: "What, Why and How" – Software Evolution – The Next Step
Part 4: Big Data and Semantic Tools at work
Part 5: Semantic Toolbox and its Magic for Validation of Development
The message from 2040 | **Discussions with the first readers** | **Buy the book**

Short extracts from Part 1: Knowledge Driven Architecture

Table of Contents

Business and technical people don't always understand each other

That might be an understatement.
While technology people speak XML and Web Services, business people prefer natural language.
Translating from the language of business to that of technology is called the "development process."
"Cooking" an application involves several teams and layers of translation.

Even after we have developed multiple services and promoted Service-Oriented Architecture (SOA), after we've invested in the enterprise service bus, business modeling tools, and more, we continue to have to jump through the same hoops.

Don't get me wrong. I support the use of SOA. It saves on budgets and the bigger the company the bigger the savings.

But to fully benefit from SOA, we need to give subject matter experts the keys to Business Architecture.

This will further the convergence of Business and Technology, preventing the usual "lost in translation" and "too-late-for-market" effects.

IT is Not about Infrastructure

Doing "more with less," and continuously "rightsizing," (shareholders like the sound of these) put more pressure on IT.

More than 50% of IT budgets go to "keeping the lights on," that is, maintaining infrastructure. Inherited from the past and obtained through mergers and acquisitions, multiple systems from many vendors need a good eye and a skillful hand to keep them running 24x7. Each new project adds more: more databases with data tables, more components and complex connectivity.

At the same time, each application focuses on specific pieces of information to display data for a specific audience under specific rules to answer specific business questions.

The point of that long and meandering sentence above is that the real focus should be on a big picture, not just the small pieces, and more on information than on infrastructure!

Focus on Information

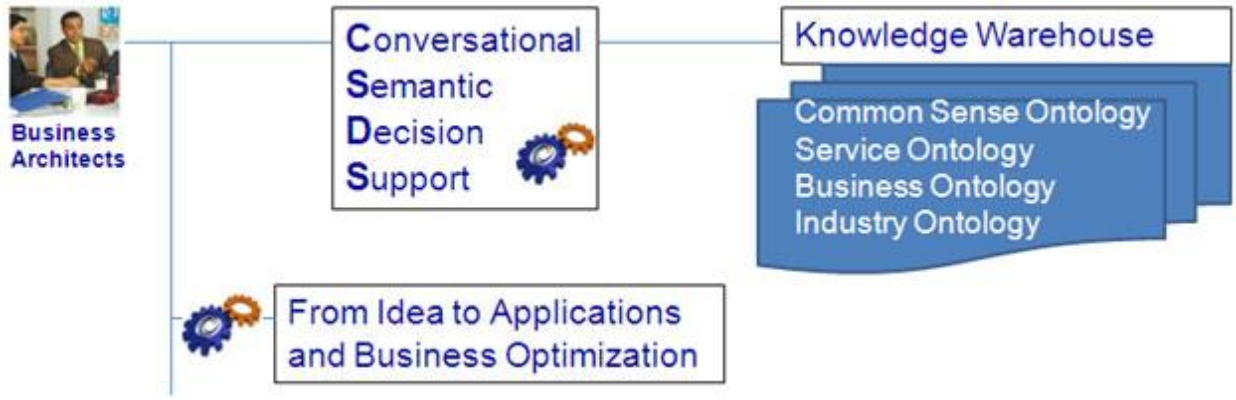
What is the alternative? Let's use a food analogy: "What do you do if you are hungry and cooking is not your sport?" You go places where food is prepared for you. You go out or maybe to your kitchen, if you're lucky enough to have a spouse willing to make you dinner.

How does this analogy apply to application development? Imagine that all the information related to your company is already collected in a single database and "ready to eat." A single service that creates a query for this database can understand your language and translate your request into a proper query or product order that is also an informational transaction in the same database, "sending in your order," for the information you want to "eat." And if you don't know the terms "on the menu," the service will initiate a conversation to help you. For example, a Business Analyst (BA) writes a line of requirements: "application starts with login."

The service would reply "Do you mean the Authentication Service?"

The BA would confirm and the service would ask "What Roles and Privileges do you have in mind for your users?"

This friendly dialogue would create a working application on-the-fly.



How realistic is this? A NoSQL databases like Triple Store [1] is already a reality. They are available from multiple vendors and some can be downloaded for free. The benefit of this type of storage is in its simplicity. Triple Store does not require data modeling or new tables. Think about a single table of unlimited size, with three columns. Any database administrator would tell us that this is not efficient. The administrator might be right for a single specific task, but this is extremely efficient for multiple tasks dealing with terabytes of corporate information. For example, the DoD plans to transition from relational databases to NoSQL storage and re-allocate a significant part of its future IT budget from maintenance to development [2].

The conversational approach was described in my previous book on integrated software and knowledge engineering, "Integration-ready Architecture and Design [3]." The Conversational Semantic Decision Support (CSDS) section, which is coming later in this book, expands on that

subject. A simple example of an application with CSDS is conversational search. Still, the trickiest part is collecting complete information encompassing a company and its business.

Information: Structured, Unstructured and "Tribal Knowledge"

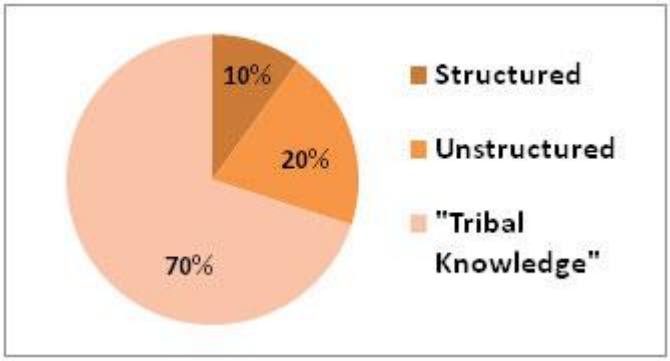
What is so hard about collecting all the information related to a company and its business?

There are two major parts to electronically stored information: *structured* data and *unstructured* data.

Structured data are very formal. They are defined in terms so precise that even computers can easily understand them. Databases, Business and Data Models, Services and XML Files are structured and understood by computers.

Unstructured data are documents and communications artifacts, like taped messages and video clips that make sense to people. The knowledge captured in unstructured data is generally unintelligible to computer systems. We'd like to change this. We'd like to use this knowledge to make computers a bit smarter. Semantic technologies help us to better organize unstructured information and create a conceptual model (ontology) of the knowledge that is represented there. More on semantic technologies will follow a bit later.

The majority of the information used daily in the routine of a business has never been captured. It is so-called "Tribal Knowledge." My conservative estimate of the ratio between structured, unstructured and "tribal" knowledge is 10%, 20% and 70%. For many companies, more realistic numbers are 5%, 15% and 80%.



- Informational gaps in "tribal knowledge" rarely noticed by a single subject matter expert (SME).
- The problem becomes more visible when several SMEs participate in a complex process.
- We cover informational gaps in every field with multiple meetings and phone calls.

Constant (and Expensive!) re-discovery is our "Normal" Process.

It becomes abnormal after we have invested in SOA and would like to link business vision to services. We feel the growing dissonance between advanced architecture and old knowledge culture.

Looking for a black cat in a dark room

In the corporate world, each clerk and department has their own knowledge compartment.

Prepared for consumption by an author or a single group, information is based on "tribal knowledge" assumptions and naturally has multiple gaps, especially for other groups.

In increasingly interconnected businesses, informational gaps lead to productivity loss.

Compartmentalized information is usually hidden and locked inside complex tools. No surprise that we spend from 30 to 50% of our time looking for information. Not because we love searching… It's just hard to find something that was hidden (not intentionally!) and especially something that has never been captured.

We often find ourselves looking for a black cat in a dark room.

When looking for a black cat in a dark room, the first thing to do is to turn on the light. In our case, this means to unlock data, which are hidden behind complex tools, and to improve data organization and taxonomy, effectively creating Transparent Enterprise. IT Transparency is one of the most important corporate goals according to Gartner Research [4]. Transparent Enterprise helps to find captured information and reveals informational gaps.

The real fun starts when we can clearly see that the cat is not there…

Changing corporate knowledge culture and finding the cat

To fill in the informational gaps, often in the "tribal knowledge" territory, we'll need both technical and organizational frameworks.

Collaboration of business and IT under an Information Governance umbrella is absolutely crucial in this effort. Information Governance (a natural extension of Data Governance) will go beyond the structured data, to promote common language across the enterprise, and to guide the process of creating different kinds of content including development documents, business

rules and scenarios. Information Governance will facilitate the adaption of policies and methods offered by technical frameworks.

The technical frameworks include semantic information architecture with integrated structured and unstructured data. The key to such integration is a conceptual model (ontology) of a business organization and processes.

Here is the Catch-22. The conceptual model must be based on "good quality" information with no informational gaps. One approach is investing upfront into building ontology. Usually, this involves some consulting help from a knowledge engineering firm. Consultants would interview employees, and turn what they learn into a limited ontology model. This expensive but necessary help can get the process started but is not ideal. Generally speaking, managing a company's information should be one of the **core** competency areas of a company, not a subject for outsourcing. The biggest problem we face today is how to engage a company's SMEs in the process of sharing knowledge and how to leverage this engagement to fill in informational gaps and complete the model [5]. SMEs are more than busy with their daily routine and will not change their ways… unless you offer immediate benefits.

Conversational Semantic Decision Support (CSDS) helps to solve this problem. CSDS is rather a method than a tool to gradually engage a SME by providing immediate help and increasing efficiency in whatever work is executed daily by a SME. The main tasks performed by CSDS are:

- Helping a SME with the right questions: providing a conversational script with the direct hints to capture important information.
- Enabling good answers with semantic support and checking the SME's entries against Data and Business Models and growing ontology.
- Establishing a process where the SME's interactions with the program not only help the SME but also contribute to building a knowledge warehouse.

I've just read this last paragraph. It sounds so dry… although we are talking about very exciting changes in the nearest future. Hopefully, the message below might serve as a good illustration to what is coming.

Read in the book how to step by step improve the process …

The message from 2040

I participated in a small trekking expedition to Everest North Face, advanced camp, 6500 meters. *Follow this link for more information about this adventure…* *http://everest6500.com* *Returning back to civilization I found many "how r u" messages.* *They were handled with the simple "alive" answer.*	

Then, there was this strange message.

The message came from the address autodistribution@2100archive.fu.

The subject was "Unclassified: Based on events of 2040".

The content included a very brief message and an attachment.

"The attached information was extracted from the Complete Archive of Events 2000 - 2100.
The analyzer system associated attached notes with several names including yours and prepared for auto-distribution. Be aware that classified parts of information were removed from the message."

Highly intrigued, I opened the attachment and started reading. My impression was that both, the beginning and end as well as some pieces in the middle had been cut, most probably as classified information. Nevertheless, the story was relevant to my work, confirming some of my ideas and expanding others.

Based on events of 2040: Psycho…

The presentation was over with no happy ending.
"-Wait a minute. This is unbelievable! Did you show this report to anyone else?"
"-No, you will be the first reader."
I looked straight at the presenter.
"What was your original task?"
"-You can call me Psycho. Many people call me Psycho. It became my nickname.
My task was to research and report the reasons behind falling productivity on three major robot plants, so called modeling factories."
"-And you think…"
"These are objective measures and the results of comprehensive conversational tests."
"-Who conducted the tests?"
"-I did. This is my specialty."
"And they admit…"
"No, there has been no intentional sabotage. No one understands why their performance indicators are lower. They do not have enough data to see the big picture. And some of them show significantly increased performance. But this is a minor deviation from the major trend. "

"So, it is just your personal conclusion?" I was not happy that I asked that question, but it was too late.
"-The conclusion is based on the six-month research and statistical analysis. "
"-What do you suggest?" It was against the rules, but I wanted the answer.
"-You know, you cannot use my recommendations. At this point I have none. More questions?"
He was right. There were many rules against questioning a reporter. Especially in this case: the reporter was not a human. But I did not expose my knowledge and continued to question my visitor.

"No… Except one: who gave you this assignment?"
There was a minute delay – "This person is no longer with the company and I cannot access his name and other personal data."
That was somewhat strange. People freely entered and left the company, and their data usually were available with rare exceptions.
"Thank you! I think we are done here."
His image immediately disappeared clearing a beautiful view of the mountains. It was Everest North Face. The giant pyramid looked steep and proud on this unusually sunny day.
I opened the report. The title was "Collaborative conflicts in robot teams."

Semantic terms: the difference between Ontology and Taxonomy …

I'd like to step back and elaborate on several terms and ideas that deserve explanations and examples.

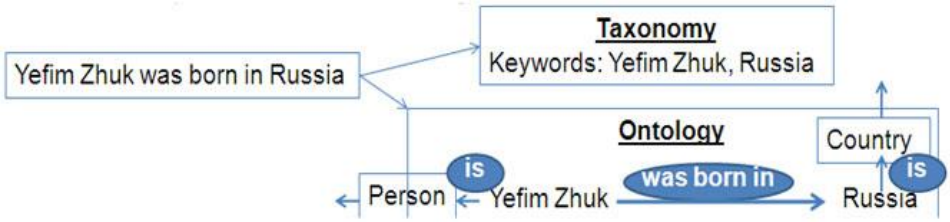
If you agree with Sir Francis Bacon that "Knowledge is Power", it might be no surprise for you that there is a science of handling and computerizing knowledge. It is called Ontology or Knowledge Engineering. The word "Ontology" also means a collection of related facts and terms.

Semantics is the study of meaning. While processing information, a semantic system focuses on understanding the meaning of that information.

This is a good point to discuss the difference between Taxonomy and Ontology.

Taxonomy collects keywords to describe content. Ontology uses more powerful methods to create more detailed meta-data models. In addition to collecting the key vocabulary, ontology picks up on the relationships between the keywords effectively building a semantically rich and much more meaningful model of the content.

Let's say we have a sentence "Yefim Zhuk was born in Russia." Taxonomy would only use two keywords from the sentence, the name and the place. Ontology would also include the relationship between the two keywords and create a graph representing the content in a greater level of detail. This graph can be extended and later used for querying, in other words, for asking questions related to this information, such as "what is Russia". . . .



By being more formal and detail oriented, ontology helps us elevate information to the next level understood by computers. Of course, we must deal with high quality Information ("the cat is there").

Multiple ontology modules represent multiple knowledge domains and views.

Integration of modules and resolving possible conflicts can be done via ontology-maps.

Ontology modules can be built gradually in the daily interactions of SMEs with the computer program, Conversational Semantic Decision Support (CSDS). Each conversation will work in both directions: helping the SME and contributing to the knowledge modules with SME entries.

Transitioning From "What" to "How" and explaining Conversational Semantic Decision Support (CSDS)

Let's take a closer look at a possible conversation between a SME and a machine.

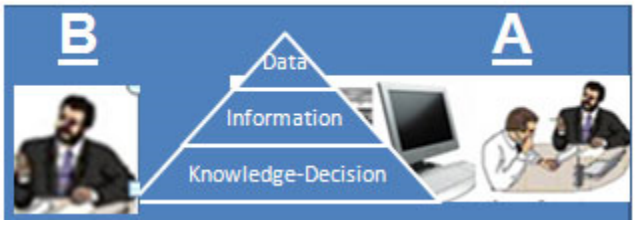
1. A Business Analyst (BA) writes a line of requirements: "application starts with login" and the service would reply "Do you mean the Authentication Service?"

CSDS consumes a line entered by the BA and uses a semantic model to map the line to one of the existing service names. In this case it was the Authentication Service. CSDS will ask for a confirmation, following the rule: *a decision has to be made by a person*, not a machine, while the machine performs all the boring part of work, like scanning against models and catalogs and mapping to proper terms and formats.

2. The BA confirms and the service asks "What User Roles and Privileges do you have in mind?"

The machine started a conversation to retrieve important information that otherwise can be omitted. Of course, this conversational script was prepared by someone upfront.

While Business (a SME) knows the answers, an Architect, trained with a systems approach, can provide good questions for the conversational scripts.



An important component is collaboration between a business SME and an architect.
The bridge between Business and Architecture helps us to develop the field and skills of Business Architecture.

3. The SME's answers are checked against the terms of Data Dictionary, Business Model and growing ontology. This "Reality check" and the following feedback to a SME help fill in the informational gaps. If the BA introduces a new Role that is not known to the model, CSDS would ask to define the role via known properties.

This is not an artificial intelligence (yet) but a good combination of a computer's power with people's expertise.

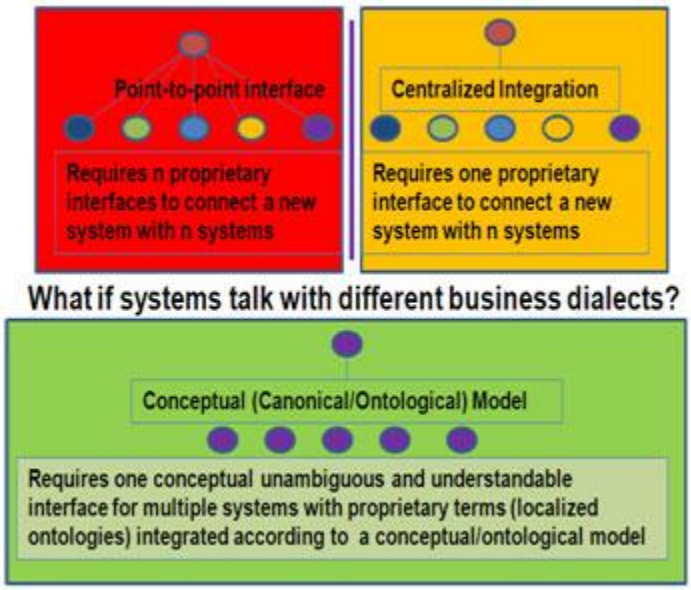
Every step should provide more help to a SME, which in turn will help grow the tree of knowledge and related productivity. This is a natural result of daily interactions between a SME and the system. Here are several examples, practical business cases where CSDS will shine:

a) Formalization of Business Rules

One of the current development trends is a shift from built-in code business logic to rule-based applications. Since they are more flexible and quickly adaptive to business changes, rule-based applications live a longer life and provide higher return on investment. Conversational semantic decision support can be very helpful in collecting and formalizing the rules [5]. CSDS will make sure that the rules are expressed in known terms and rules criteria are directly tied to existing data.

b) Service integration and data mapping

In its evolution, service integration has developed from Point-to-point interfaces to the Centralized Integration Design Pattern and then to the Canonical Interface Design Pattern.



Service integration between different companies usually adds another dimension to regular challenges: different terms and business dialects.

The Canonical (Conceptual/Ontological) Integration Model expands the Centralized Integration design pattern and bridges the terminology gaps.

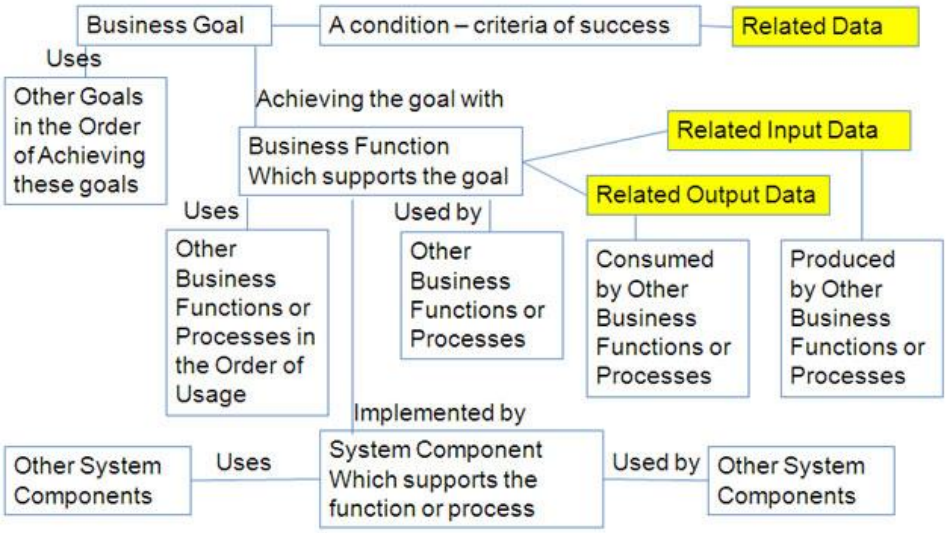
Interaction with CSDS allows a Business Analyst to capture the results of analysis in a format readable by people as well as understandable by a computer and ready for auto-validation.

In this case, the results of analysis are automatically transformed into a decision table to drive service mediation at run-time and also become part of the “localized” ontology that provides the mapping between the conceptual ontological model to proprietary models and values of legacy systems.

c) Enabling best practices in development

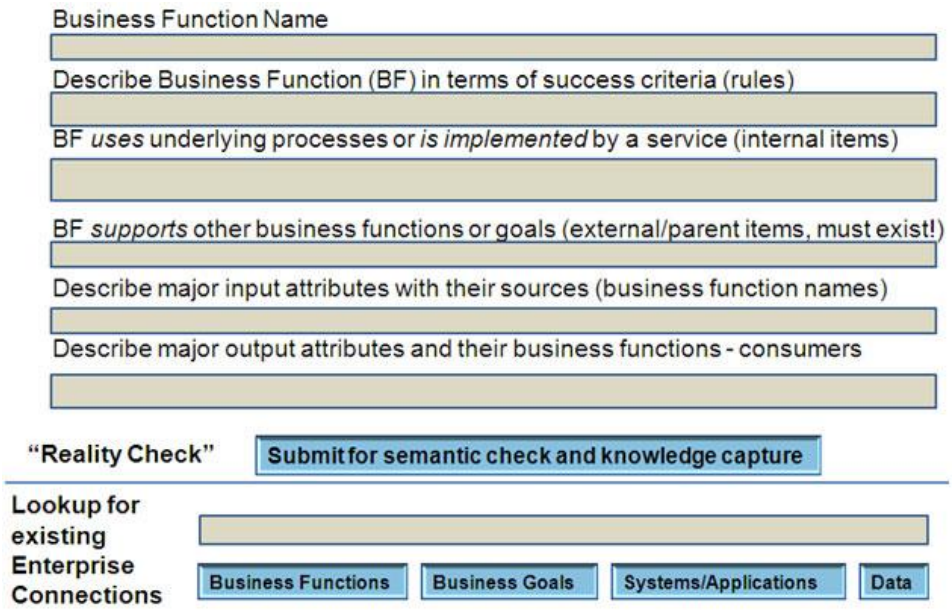
Not just teaching best practices, but in a very practical way helping their implementation.

Let us start with requirements. Business Analysts traditionally write requirements in MS Word documents with a mix of “what” and “how”, with both: informational gaps and unnecessary details. This old and comforting practice on the business side causes extra time on the development side for re-discovery and translation layers. We can effectively optimize the development process from requirements to implementation by using Conversational Semantic Decision Support. We can retrieve necessary information from business users by asking the right questions. The architecture diagram below serves us as the base for creating conversational scripts or forms.



We go far beyond drawing diagrams. We actually enable best practices and decrease information gaps.

The simplest example of the Conversational Support is a form, like the one below.



The form will retrieve initial information and check information against a growing ontology. The system's feedback will include hints to help a user providing valid entries. For example, it will not accept unknown names in the parent field (BF supports other business functions or goals), and will re-enforce a hierarchy with no gaps. At the same time, internal items can include new names and the user will be invited to define them later.

Exciting Opportunities

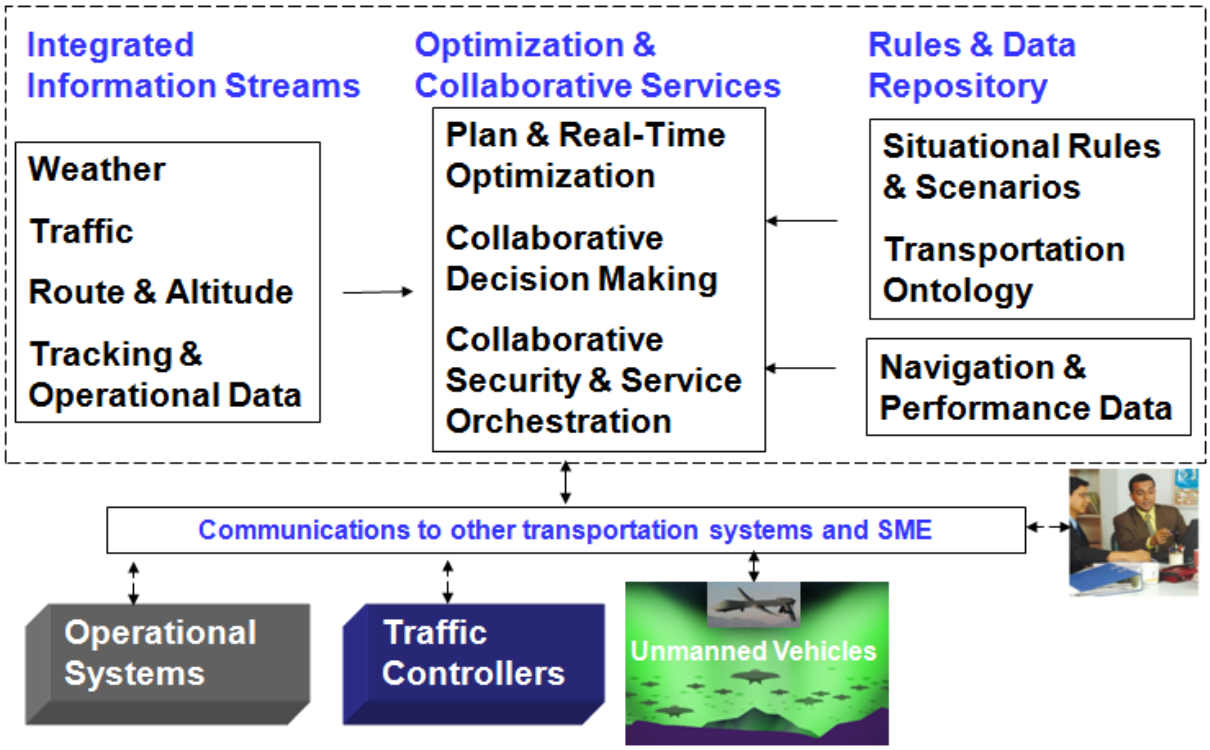
I have to admit that I've simplified several things. As Albert Einstein said, "Make everything as simple as possible, but not simpler." And then he wrote his "Unified Field Theory" and everything that was written later on this subject was even more complicated.

One thing is clear: with the volume of information doubling every year, and with increasingly interconnected departments and corporations, old technology is on its way out (this could be quite a long way). Semantic technology, the cool new kid on the block who also happens to be pretty darn smart, is on its way in.

In the future, a new class called Knowledge Engineering will be introduced in every school along with the subject of Critical Thinking. Modeling tools that have Business and Development views today will add an Ontology view tab to the front page. This is happening as you read these lines.

The conversational approach is already adapted in its simplest form by many existing tools. Semantic support of the conversation is a very logical and powerful step ahead. We will be able to gradually decrease infrastructure expenses, close some boring chapters of the IT story, and open the door for new exciting opportunities.

Imagine multiple data streams intercepted by multiple run-time systems. Multiple services working in collaboration and changing their behavior based on situational rules and scenarios...



System security has traditionally been based on the security of individual services. Much more powerful protection can be provided by considering the services in their collaborative work. A beautiful idea of collaborative security and decision making is turning into a working system [14] accessible by us, humans, and them, computer systems and services.

Another invention, which is also based on knowledge-driven architecture, is adaptive robotic systems that can learn by conversing with people and store new skills as orchestrations of services [15].

A fundamental problem of current robotics and overall software applications is related to multiple translations from natural language of task requirements to compiled and integrated working systems. Current robots are programmed to perform relatively simple, well defined and predictable tasks.

Read more in the book…

Based on events of 2040: The Report…

The report started with the history. Designing and discussing robots were hot subjects for decades. The discussion about human – robot collaboration and wars brought high hopes to some and scared the others. Meanwhile, the robot skills steadily expanded human capabilities, especially in environments hostile to humans. In most cases, people and robots existed in different dimensions with no ground for any possible conflicts. Nevertheless, every robot action was always checked against the "good for people" rules.

Overtime, the dust settled, giving way to more practical questions. Should we invest in universal "super" robots or combine specialized robots into teams? Designing robots for multiple areas, providing connections to the knowledge domains, and engaging in the conversational study with the best world minds seems to be necessary, although long and expensive.

Very soon the robots learned how to help us in the education and testing processes. They actually created new test sets and compared these with existing ones. They improved educational materials and provided more individual skill-based layers. They introduced better evaluation instruments that looked more like games than tests. These games helped to precisely measure a student's disconnect at each point of study. Robots constantly learned and tested themselves. They never cheated. Some people said that robots cannot lie and others said that robots lack imagination.

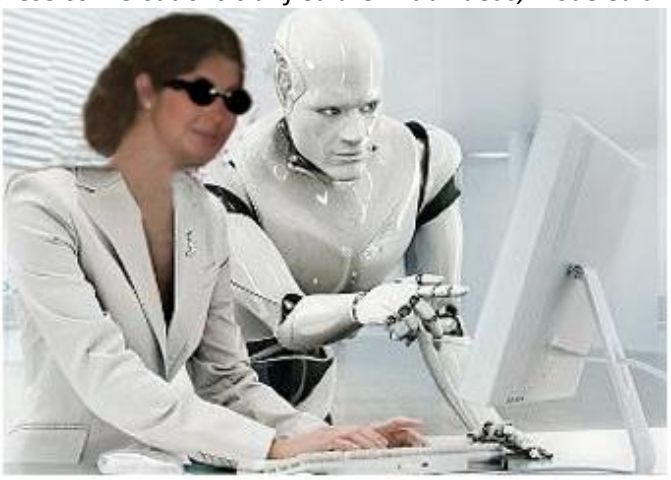
Most "super" robots became teachers, passing their knowledge to other robots and also teaching in the regular virtual schools. They were extremely successful with children. Their enormous knowledge and quick thinking, their ability to replicate their custom copies while gaming simultaneously with multiple participants, and most importantly their lack of emotion contributed to their success. Robots kept their cool in the kinds of situations that would drive a human teacher crazy.

Verbal interactions and situational games facilitated by robots helped us solve at least two problems. Parents had their break when they needed it, while children still enjoyed undivided attention in a highly educational and comfortable environment. Kids loved it and the price was right too. This was the first opportunity to start early childhood education for the whole planet with a coordinated cultural program

under the UNESCO umbrella. UNESCO subsidized the program with the "Peace for Our Children" fund. Very few people knew about the non-stop internal fight between the program donors, Localists, who pushed for local inclusions and Globalists who insisted on complete unification of the curriculum. Nevertheless, the program was instrumental in increasing knowledge and understanding among the world's youth as the best protection against violence and wars.

But education was the only area where super-robots made sense. In all other areas they were too expensive for what they were doing and their payoff or "return on investment" was very questionable. Creating specialized robots and their growing teams became the mainstream of robot development.

Robots were made from materials and in shapes which fit their purposes best. The modeling factories which designed and built custom robots and teams grew into the largest industry. After starting in the technologically advanced countries, these factories slowly spread to the more remote parts of the world, repeating the usual process of maturing technology. Modeling factories grew out of 3D printing, as people first called it. From replication or printing, they moved to modeling and creating new things, streamlining the manufacturing process and making it available to the average person. What we used to call design and development was transformed into a direct conversation between a person and a modeling factory. Initiated by a person and lead by globally connected specialized computer systems, these conversations clarified the initial ideas, modeled and manufactured the desired product.



Large modeling factories produced smaller modeling factories, highly customized for the clubs, restaurants, and private homes.

For example, almost every woman had a custom modeling factory that looked like a big old mirror, but produced clothes that fit her style and size.

Of course, it was necessary to upgrade or change these factories to catch up with the fashion changes.

Men focused on modeling food, drink, and transportation. Fancy robot plane with a mix of bio- and nano-engineering, could be created by individuals, but belonged to the clubs, which took care of safety as well as other aspects of life in transition.

Federal law highly restricted the usage of man driven vehicles (MDVs) to remove human error out of the transportation equation. This encouraged new travel plans and models. For example, a submarine combined with a flying saucer became a perfect vehicle to explore deep spots of Pacific Ocean and Himalayan heights in a week-long tour.

The three biggest modeling factories were located in the Sahara desert. Thousands of robot teams consumed local silicon, sun, and natural bio resources, transforming this complex mix into final products. The modeling factories were connected and shared their specialized skills and knowledge. Some of them had access to and could occasionally call subject matter experts. Sharing was never an easy procedure, sometimes involving multiple companies in dynamic negotiations over the features, time, and resources. Of course, it lasted milliseconds and was completely hidden from the end users.

Over the years, the productivity of the factories steadily grew ... till the last year. Last year all three biggest factories seemed to slow down their progress. It was not very noticeable, but apparently some highly positioned executive at our company ordered a special investigation. This unknown person left the company and the name mysteriously disappeared. Anyway, this investigation was assigned to a specially designed "super" robot with the ability to collect and analyze all communications between the company's robots during the year. The super robot was also highly skilled in the relatively new field of robotic conversational psychology and actively used its methods during the analysis.

The report landed in my hands. There was no way to avoid this subject at the next management meeting, although I felt very uncomfortable and unprepared bringing this bomb to the meeting room.

I looked at the window. There was a minute of clearing when the summit of Everest came out of the clouds. The goddess mother of Earth, Qomolangma, as Tibetans and Nepalese call the mountain, was not in a mood.

Read more in the book…

Part 1: Knowledge Driven Architecture | **Part 2: Transitioning to Semantic Cloud**
Part 3: After SOA: "What, Why and How" – Software Evolution – The Next Step
Part 4: Big Data and Semantic Tools at work
Part 5: Semantic Toolbox and its Magic for Validation of Development
The message from 2040 | **Discussions with the first readers** | **Buy the book**
